The University Of Sheffield.

Effective, Usable and Learnable Semantic Search

Khadija Mohamed Elbedweihy

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy at
the University of Sheffield
Department of Computer Science

April 2, 2014

Abstract

The exploitation of the underlying semantics of data inherent in the vision of the Semantic Web tackles the limitations of the traditional keywords-based retrieval model and has the ability to change the way search is done. The proliferation of Open Data published on the Web in recent years has driven significant research and development in search. As a result, there is a wide range of approaches with respect to the style of input, the underlying search mechanisms and the manner in which results are presented. Although the performance or effectiveness of these approaches is usually evaluated, understanding their usability and suitability for end users' needs and preferences has been largely overlooked. This is the main motivation behind the work presented in this thesis.

The thesis, thus, presents different pieces of work in this area. The first part focuses on investigating the usability of different query approaches from the perspective of expert and casual users through a user-based study. The findings of this study show the strengths of graph-based approaches in supporting users during query formulation with a drawback of high query input time and user effort. Therefore, in another user-based study, learnability of a graph-based approach is evaluated to assess the effects of learning and frequency of use on users' proficiency and satisfaction. The results of both studies suggest that the combination of a graph- based approach with a NL input feature could provide high level of support and satisfaction for users during query formulation. This is, hence, the third piece of work presented in the thesis: a hybrid query approach together with a user-based evaluation to assess its usability and users' satisfaction. The thesis also presents thorough analysis of state-of-the-art in semantic search evaluations and describes a set of best practices for running them based on this analysis, lessons learnt from the Information Retrieval community and on my own experience in evaluating semantic search approaches.

Dedication

This thesis is dedicated to the memory of my only brother, Abdulrahman Elbedweihy (Bido), 28 years old, who passed away on August 16th, 2013 on May 15th Bridge in Mohandesseen residential area in Giza after courageously participating in a fight for his country's freedom and future.

After Friday prayer, Bido along with thousands of fellow Egyptians participated in "a day of anger" following Wednesday's massacre on peaceful protestors by the Egyptian Police and Army controlled by the coup leaders. He marched alongside friends, family, and other fellow Egyptians as "one". Through these peaceful marches and protests, the police stealthily hid in nearby buildings and helicopters with snipers targeting random civilians and shooting them dead to spread fear among crowds. Bido unfortunately was one of their victims. He was shot unexpectedly from the back and died on the scene. He was a martyr and a symbol for the freedom and security of Egypt, as well as a voice for the innocent victims who have been murdered as a result of the corrupt people who want to see power remain in the hands of few. Bido, along with other Egyptians, refused to stand for such injustice and corruption and stood his ground to make a difference in Egypt. Bido's death was not in vain; he represents a change, and the future.

The tragedy of Bido's murder has saddened the hearts of many family and friends. He was young with his whole future ahead of him. He worked hard to reach his goals and was very determined and ambitious to succeed in almost everything he did. He was very caring towards his friends and family, and most would describe him as intelligent, funny, and a person who would always be there to help and support others. Bido was kindhearted, warm and children loved being around him. He was an honest man who lived for peace, morals, and to genuinely do what was right in his heart. He was passionate about worldly issues and was not afraid to speak his mind about what was right. It brings many of us great sorrow to know we will not have Bido beside us growing old, sharing memories, the joys of life, sorrows, and to not ever see his warm smiling face. His death was sudden, and has left a hole in the hearts of loved ones. There was no chance to say goodbye or tell him how much he was loved and appreciated by everyone around him. Bido will always have a special place in our hearts.

Bido's last facebook message a few hours before he was killed reads:
"All of us are marching tomorrow (to protest the killing of innocents).. We are peaceful not armed..We have hearts full of hope for a new future and we are not afraid of bullets. Our bare chests are stronger than your tanks.. We are not afraid and we will not run.. Some of us will die but we know that freedom is costly. We will never be slaves to the

army and to the police again (that is run by the coup leader)."

Finally, I also dedicate this thesis to the faithful thousands who have been killed in Rabaa, El-Nahda, Ramsis and all other places, defending their choice and freedom in August, 2013. To those who defend freedom, anywhere.

Acknowledgements

"*March on. Do not tarry. To go forward is to move toward perfection. March on, and fear not the thorns, or the sharp stones on life's path*". – Khalil Gibran

First and foremost, all praises to Allah for his blessings that enabled me to complete this thesis. My gratitude and deep appreciation then goes to my supervisor Prof. Fabio Ciravegna for his guidance and support over the years spent to pursue this work. I must thank him for giving me this great opportunity to join his research group and pursue my studies under his supervision. His continuous efforts in transferring to me his knowledge on conducting research, reviewing others work and writing have been invaluable. He always challenged my ideas, which, although stressing, provided enormous improvements to this thesis and to my understanding and ability to explain my work.

In these years, I have had the chance to work with a very special group of colleagues at the Organisations, Information and Knowledge Group. I would like to thank them all for many interesting discussions, and for the time they have taken to participate in my usability studies. I like to specially thank Elizabeth Cano, Suvodeep Mazumdar, and Andrea Varga who have been amazing companions and office mates. I must also acknowledge and deeply thank Stuart Wrigley, my mentor, for always finding the time for me. Without his advice, constructive critique and feedback on different aspects of the thesis, it would not be in this shape. I enjoyed each of our conversations and the work we have done together, and I have truly learnt so much from him. Finally, I am thankful for many others who helped me along the way, including Victoria Uren, Aba-Sah Dadzie and Paul Clough.

I am truly indebted to my family for unconditionally supporting me, both financially and emotionally. I am exceptionally grateful to my father – my idol – the man who always believed in me and in my ability to "do it". He has always been there for me during this long journey with his motivating words and actions. In the toughest of times, remembering him and his sayings to me – "And so it's not wishful thinking that grants you that which you long for; life's aspirations are rather contested" – gave me the strength to continue. To my mother, I cannot find enough words to express my love and gratitude. Without your constant prayers and care bestowed upon me, I would not have been able to reach this point. Thanks are also due to my sisters Safiya, Sarah and Alaa, my lovely nephews and niece who brought laughter to my days, and to my best friend Ayat.

Last but not least, I must express my profound gratitude and eternal love for my devoted husband, Mohamed. Being able to work steadily and finish my PhD (marathon) would have never been possible without his support, encouragement, and most of all, inspiration. Throughout all the difficulties I have faced during these years, his motivating words, consolatory hugs and great sacrifices kept me going. I will always remember our long conversations with his positive criticism and challenges for my work which always led to invaluable suggestions and interesting ideas.

Publications

The work presented within this thesis (covered in the given chapters) is published in the following papers:

- NL-Graphs: A Hybrid Approach toward Interactively Querying Semantic Data. K. Elbedweihy, S.Mazumdar, S.N. Wrigley, and F. Ciravegna. The 11th European Semantic Web Conference (ESWC2014), Crete, Greece. (2014) : *Chapter 9.*
- An Overview of Semantic Search Evaluation Initiatives. Khadija M. Elbedweihy, Stuart N. Wrigley, Paul Clough, and Fabio Ciravegna. (Revisions submitted to the Journal of Web Semantics, January 2014) : *Chapters 4, 5, and 10.*
- Semantic Search Approaches: State-of-the-Art Survey. Khadija Elbedweihy, Stuart N. Wrigley, Victoria Uren, and Fabio Ciravegna. (Submitted to the Knowledge Engineering Review, February 2014) : *Chapter 3.*
- Using BabelNet in bridging the gap between natural language queries and linked data concepts. K. Elbedweihy, S.N. Wrigley, F. Ciravegna, and Z. Zhang. NLP & DBpedia Workshop, 12th International Semantic Web Conference (ISWC2013), Sydney, Australia. (2013) : *Chapter 9.*
- Affective Graphs: The Visual Appeal of Linked Data. S.Mazumdar, D.Petrelli, K. Elbedweihy, V. Lanfranchi and F. Ciravegna. Semantic web : interoperability, usability, applicability. (2013) : *Chapters 7 and 8.*
- Evaluating Semantic Search Query Approaches with Expert and Casual Users. K. Elbedweihy, S.N. Wrigley, F. Ciravegna. Evaluations and Experiments Track, 11th International Semantic Web Conference (ISWC2012), Boston, USA. (2012) : *Chapter 7.*
- Improving Semantic Search using Query Logs Analysis. K. Elbedweihy, S.N. Wrigley, F. Ciravegna. Interacting with Linked Data (ILD 2012) Workshop. Located at the 9th Extended Semantic Web Conference, Heraklion, Crete, Greece. (2012) : *Chapter 11.*
- Identifying Information Needs by Modelling Collective Query Patterns. K. Elbedweihy, S. Mazumdar, A. E. Cano, S. N. Wrigley and F. Ciravegna. Consuming Linked Data Workshop (COLD 2011). Located at the 10th International Semantic Web Conference, Bonn, Germany. Proc. CEUR-WS.org Vol-782. (2011) : *Chapter 11.*

- SEMLEX - A Framework for Visually Exploring Semantic Query Log Analysis. S. Mazumdar, K. Elbedweihy, A. E. Cano, S. N. Wrigley and F. Ciravegna. Poster and Demonstration Session, 10th International Semantic Web Conference, Bonn, Germany. (2011) : *Chapter 11*.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

The movement from the 'web of documents' towards structured and linked data has made significant progress in recent years. This can be witnessed by the continued increase in the amount of structured data available on the Web, as well as the work done by the W3C Semantic Web Education and Outreach (SWEO) Interest Group's community project *Linking Open Data*[1] in linking various open datasets. This has provided tremendous opportunities for changing the way search is performed, and there have been numerous efforts on exploiting these opportunities in finding answers to a vast range of users' queries.

Currently, to take advantage of these opportunities and make use of this data in finding answers for their information needs, users depend on one or more of the following: 1) traditional search engines such as Google which have started to incorporate semantics into their search process; 2) Semantic Web search engines such as Sindice [TOD07]; 3) natural language interfaces such as PowerAqua [LMU06]; 4) view-based interfaces allowing users to explore the search space such as Smeagol [CD11]; and 5) mashups providing rich descriptions about Semantic Web objects such as Sig.ma [TCC+10].

Knowing this, researchers continue to evaluate these systems and approaches to identify their strengths and weaknesses in an attempt to improve their performance. Unfortunately, most of these evaluations focus on assessing the effectiveness and efficiency of these approaches with respect to various aspects such as retrieval, response time or ability to support specific types of queries. Understanding users' needs and requirements and whether they match with these approaches does not yet receive equal attention. Addressing this question is very important in order to develop future query approaches that cater to the preferences and needs of the target users.

We note that it is yet difficult to find a unified definition for *semantic search*. It has been used by different research communities including Information Retrieval, Natural Language Processing and the Semantic Web (SW) to describe different approaches and strategies employed to improve search performance and user experience. Even within

[1] http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

the Semantic Web community it could be used to refer to a broad range of applications and services. These include the ones mentioned above (SW search engines, NL and view-based query approaches targeting end users, mashups); SW browsers, such as Tabulator [BLCC+06], allowing interactive exploration and navigation of semantic data as well as other similar applications and tools more tailored to exploratory search tasks (e.g., LOD Live[2], tFacet [BH11], Aemoo [MND+12]) or very specific querying tasks such as finding relations associated to certain concepts or entities (e.g., RelFinder [HHL+09]).

The scope of the work done in this thesis is limited to semantic search query approaches targeting end users (as opposed to applications) performing simple as well as complex queries against semantic data.

1.2 Research Questions

The work explored in this thesis is motivated by the problem statement and need described above. Research on semantic search is directed to one or more of the following aspects/stages: 1) input query approach; 2) underlying search mechanisms and techniques; and 3) results presentation. The thesis reviews state-of-the-art (SOA) semantic search systems from each of these aspects and provides recommendations for future development and research covering these aspects. The main focus and most of the work done is on the first stage, *input query approach*. Therefore, the main research question investigated within this thesis is the following:

How can we design a semantic search query approach that is usable and effective beyond current state of the art approaches?

To answer this broad question, it has been divided into several questions as shown below. The work done on these focused and more specific questions facilitates the investigation of the main research question.

1. *How do casual and expert users perceive the usability of different semantic search query approaches/formats?*
2. *Can training and frequency of use of a query approach improve the proficiency level and efficiency of users (in terms of time and effort) in answering search tasks of different complexity?*
3. *Using the outcomes of the above studies, can we design a user-oriented query approach which balances usability with effectiveness and efficiency while querying complex domains?*

1.3 Contributions

The work presented throughout this thesis makes practical as well as scientific contributions to the area of semantic search, specifically trying to bridge the gap between

[2] http://en.lodlive.it/

users and systems to improve the usability and support of query approaches. These contributions are as follows:

- A comprehensive survey of SOA in semantic search. Semantic search is still in its infancy and many of the current approaches are facing challenges. This survey attempts to provide an understanding of the strengths of the different approaches as well as the challenges facing them, which is necessary for further progress and improvement.

- The usability study conducted to answer the first research question provides direct comparison of the different query approaches and a first-time understanding and comparison of how expert and casual users perceive the usability of these approaches. I believe the results and findings of this study provide a contribution for the Semantic Web community, especially for developers of future query approaches and similar user interfaces who have to cater for users with different preferences and needs.

- The user-based study conducted to answer the second research question is the first work to investigate and address learnability of semantic search query approaches and how it influences effectiveness, proficiency and satisfaction. Measuring usability in a one-time evaluation may not be sufficient for assessing user satisfaction with different query approaches. This is because the use of some systems employing these approaches is expected to require an amount of learning, and therefore assessing learnability would be essential. Both studies (this one and the above) also raise the need within the semantic search community to move towards more comprehensive views of semantic search evaluations by addressing these important criteria (usability and learnability), which are as important as assessing retrieval performance.

- A survey of SOA in semantic search evaluations, together with lessons learnt and best practices for designing these evaluations. The survey highlights the most important limitations and missing aspects in these evaluations. The best practices attempt to support the semantic search community in tackling these limitations and filling the identified gaps. They are based on learning from the IR community and also on my own experience of running semantic search evaluations (discussed above). Evaluation is highly important for designing, developing and maintaining effective systems or interfaces since it allows quantifying and measuring their success in their intended tasks. This survey and set of lessons and practices are therefore important in fostering research and development in this area.

- The work done to answer the third research question produced a query approach based on the findings drawn from the usability studies. A hybrid approach is developed 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 by adding a *NL component/feature*. Designing semantic search approaches based on careful analysis and understanding of users' requirements

and needs, rather than the designer's own understanding, is very important for the progress of semantic search and for reaching a wider population of users, not limited to the Semantic Web community.

- The thesis findings and recommendations with regards to the different query approaches and their usability and user satisfaction, as well as to the design of comprehensive user-oriented evaluations, provide contribution for future Semantic Web applications, in general, and semantic search ones, in particular.

- In addition to understanding users' requirements and preferences for producing better semantic search query approaches, it is important to understand their information needs as well. Therefore, a proposal is presented (as part of my future work, since it has not been evaluated yet) to make use of semantic query logs to identify information needs of users querying the Semantic Web and Linked Data. The findings of such a study are useful for researchers and developers, especially for linked data providers who would benefit from matching their data with the needs of linked data consumers. Additionally, the study provides insights into the patterns and trends inherent in user queries. In my view, this reveals potential for different Semantic Web applications such as a semantic search tool, which could benefit from having an advance knowledge of the most queried categories and the associated search patterns followed by users.

- The usability studies mentioned above have several findings, one of which is the need for more information returned with the search results to provide a richer experience and a wider understanding. Another proposal is thus presented (as part of my future work, since it has not been evaluated yet) for using the previously mentioned query logs to return more information for users with the results. This is the first proposal for using query logs to enrich results of semantic search tools. The strength of the proposed method lies in utilising query logs as a source of collaborative knowledge, able to capture perceptions of Linked Data entities and properties, and use it to select which information to show the user rather than depending on a manually (or, indeed, randomly) predefined set.

1.4 Thesis Structure

This thesis is divided into three parts, structured as follows:

1.4.1 Part I - Background

Chapter 2 is an introduction to the Semantic Web and its main concepts and technologies such as ontologies and Linked Data. It also establishes the most important terminology that will be used throughout the thesis such as *RDF, ontologies, RDF-S, OWL, SPARQL*, and *linked data*.

Chapter 3 introduces semantic search and discusses the opportunities offered by the Semantic Web for this research area; for instance, understanding the semantics of

the data and reasoning on it and integrating pieces of information from different data sources. The chapter also presents a review of state-of-the-art in semantic search. To facilitate the discussion, the review is structured around the main aspects in which semantic search approaches differ: input query format, query processing and transformation, query execution and results presentation.

Chapter 4 provides a background on evaluations in information retrieval (IR) and semantic search, and their approaches and methodologies. Then, it reviews existing semantic search evaluation initiatives with respect to important aspects such as the datasets used or the evaluation measures adopted.

Chapter 5 presents a thorough analysis of the semantic search evaluation campaigns – reviewed in the previous chapter – with respect to a number of critical aspects such as the datasets and queries used, the process of the result relevance decision, and the performance measures and how they are computed.

1.4.2 Part II - Methodology

Chapter 6 presents the aim of the thesis: the design of a user-oriented semantic search query approach beyond current SOA approaches. The gap in current approaches is therefore discussed to motivate the need for the intended approach. Then, the requirements are listed which the query approach should fulfill. Additionally, the requirements for the user-based evaluations conducted as part of the thesis are also presented. Finally, the different design choices followed in developing the query approach and running the evaluations, in order to conform to the requirements, are discussed.

Chapter 7 presents the user-based study conducted to understand how users perceive the usability of different semantic search query approaches. The methodology adopted for the usability study is described together with the dataset used and the setup of the experiment. Next, the results and analyses of comparing the four different query approaches are discussed together with the main conclusions and limitations of the work.

Chapter 8 presents the user-based study conducted to assess the learnability of a view-based query approach and how it influences the user's level of performance and perceived satisfaction. It is structured in the same way as the previous chapter: the methodology adopted is described and then, the results and analyses are discussed together with the main conclusions.

Chapter 9 presents the prototype developed as an implementation for the hybrid query approach which resulted from the outcomes of the previous studies. A review of the related work and the different ways in which the term *hybrid approach* was defined and used is first presented. Then, the requirements for the approach and the architecture of the implemented prototype are discussed. After that, illustrative scenarios are presented, showing the querying experience and the interaction between users and the interface. Finally, a user-based evaluation of the approach is described together with a discussion of the results and main findings and conclusions.

1.4.3 Part III - Conclusions

The first part of Chapter 10 summarises the main findings of the different pieces of work carried out and presented in this thesis. It explores the main research question underlying the thesis, which is concerned with developing a user-oriented semantic search query approach, and discusses the conclusions with respect to the question and the thesis attempt to answer it. Then, the second part presents a set of guidelines and recommendations for the design of semantic search evaluations. These best practices are an outcome of the analysis provided in Chapter 5, the literature from IR evaluations (discussed in Chapter 4) as well as lessons learnt by the author from evaluating semantic search approaches.

Finally, Chapter 11 discusses several ideas for future work, some of which have been implemented but not yet evaluated while others are only proposals by the author. The first is a proposal to explore users' information needs by analysing semantic query logs. A review of related work is introduced, then, the methodology of analysing query logs together with the dataset used are described. Finally, an approach for consuming the results of this analysis is discussed followed by main observations and conclusions.

The second is a proposal for using these query logs to enrich results of semantic search tools. First, the analysis performed on these logs is described together with the models created to exploit this analysis. Then, an approach to augment search results in two different ways by exploiting the generated models is discussed. Finally, a method to use these models to assist in visualising large data sets during query formulation is proposed, followed by conclusions and limitations of the work.

Part I

Background

"*Be not afraid of greatness: some are born great, some achieve greatness, and some have greatness thrust upon them.*"

– William Shakespeare

Chapter 2

The Semantic Web

2.1 Introduction

In 1999, the inventor of the World Wide Web Tim Berners-Lee had the following vision for the Semantic Web:

> "I have a dream for the Web in which computers become capable of analysing all the data on the Web ; the content, links, and transactions between people and computers. A Semantic Web, which should make this possible, has yet to emerge." [BLF08]

The Semantic Web is an expansion of the current Web which is concerned with how machines can consume and process the data on the Web. This data is currently only suitable for human consumption. Adding semantic markups, or metadata, creates an environment in which the meaning of information is explicit for processing by machines. This allows machines to help people carry out most of the laborious and time consuming tasks that must be done by humans on the current Web, such as reasoning over data and aggregating pieces of information provided by distributed sources. Additionally, the Semantic Web is about publishing data (e.g. people and places are resources), sharing, reusing and connecting it using URIs, in the same way documents are published and connected on the current Web. This would turn the Web into a global database containing information about all types of real-world entities that can be queried to find answers for various information needs.

Ontologies have been known as one of the cornerstones of the Semantic Web. The term *ontology* was historically used to describe a branch of philosophy that tries to study the nature of reality and what exists. It was later used by the Artificial Intelligence (AI) community to capture and model knowledge and allow reasoning on this knowledge. Since then, ontologies have been viewed as conceptual models that explicitly define common vocabulary used across a domain. They formally conceptualise this vocabulary in the form of concepts and relations that exist between them. This is the purpose ontologies serve in the Semantic Web: they add explicit meanings to the information, making it machine-processable.

For data and ontologies to be readable by machines for sharing and reusing, new languages and technologies were introduced, some of which were standardised by the World Wide Web Consortium (W3C)[1]. Those include but are not limited to RDF[2] as a data model, OWL[3] as an ontology language as well as SPARQL[4] as a query language for the Semantic Web. In his talks, Berners Lee referred to the Semantic Web as a *Web of data* that can be processed directly and indirectly by machines. It is thus very common for the two terms to be used interchangeably by the Semantic Web community and also throughout this thesis.

2.2 Resource Description Framework (RDF)

The Resource Description Framework (RDF) is the standard data model for representing information on the Semantic Web. It introduces a standard framework of expressing this information as well as ensuring interoperability between applications that both produce and consume machine-processable data [MM04].

RDF uses *statements* to model information – in an explicit form – about *resources* on the Web and their *properties*. Resources are entities or things of interest to us (e.g. place, person) and they are uniquely identified using URIs[5]. Properties describe traits or characteristics of resources and their relations with other resources. RDF statements are known as *triples*, since they take the form of *subject-predicate-object* expressions. An RDF statement describes a relationship, indicated by the predicate and holds between the subject and the object of the triple. The subject denotes the resource we want to talk about (e.g. "Egypt"). The predicate represents the relation between the subject and the object (e.g. "capital"). Finally, the object is the value of the predicate specific to that subject (e.g. "Cairo").

A set of RDF statements or triples form an *RDF graph*. For example, Figure 2.1 shows an RDF graph describing statements about 'Egypt'[6]. There are two subjects in this graph: 'Egypt' and 'Cairo' which are identified by the URIs <http://dbpedia.org/resource/Egypt>and <http://dbpedia.org/resource/Cairo>respectively. Similarly, predicates are identified using URIs. For example, the graph shows that Egypt is related with another resource, which is <http://dbpedia.org/resource/Egyptian_pound>, through the predicate <http://dbpedia.org/ontology/currency>. As shown in the graph, statements can be linked together to form bigger graphs about different subjects when the object of a statement (e.g. dbpedia:Cairo) is used as the subject of another.

RDF statements can be expressed in different formats. Up to the date of this writing,

[1] http://www.w3.org/

[2] http://www.w3.org/RDF/

[3] http://www.w3.org/TR/owl-features/

[4] http://www.w3.org/TR/rdf-sparql-query/

[5] A Uniform Resource Identifier (URI) is a string of characters used to identify a resource on the Internet [BLFM05].

[6] URIs are abbreviated in the graph for readability, dbpedia:http://dbpedia.org/resource/, dbo:http://dbpedia.org/ontology/, foaf:http://xmlns.com/foaf/0.1/, rdfs:http://www.w3.org/2000/01/rdf-schema#, rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#.

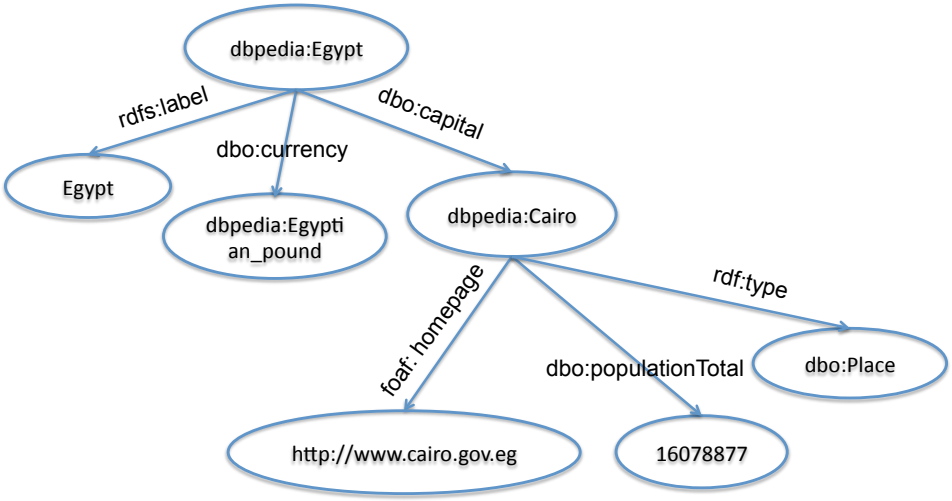


Figure 2.1: Example of an RDF graph

four different serialisation formats have been proposed, namely RDF/XML[7], Notation3 or N3[8], Turtle[9] and finally N-Triples[10]. Additionally, RDF statements can be embedded into XHTML through the use of RDFa[11], Microformats[12] or eRDF[13]. The following is part of the RDF/XML representation of the graph shown in Figure 2.1.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dbpedia="http://dbpedia.org/resource/"
  xmlns:dbo="http://dbpedia.org/ontology/">
  <dbo:Place rdf:ID="http://dbpedia.org/resource/Egypt">
     <rdfs:label>Egypt</rdfs:label>
     <dbo:currency rdf:resource="http://dbpedia.org/resource/Egyptian_pound">
     <dbo:capital rdf:resource="http://dbpedia.org/resource/Cairo">
   </dbo:Place>
</rdf:RDF>
```

Finally, information represented as RDF statements can be stored in special types of databases called triple stores (also referred to as Semantic Web databases or RDF databases). These store RDF statements into three columns: subject, predicate and

[7] http://www.w3.org/TR/REC-rdf-syntax/
[8] http://www.w3.org/DesignIssues/Notation3.html
[9] http://www.w3.org/TeamSubmission/turtle/
[10] http://www.w3.org/TR/rdf-testcases/#ntriples
[11] http://www.w3.org/TR/xhtml-rdfa-primer/
[12] http://microformats.org/
[13] http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml

object. Statements can be stored into and retrieved from a triple store using a query language such as SPARQL [PS08].

2.3 Ontologies

Semantic markup can be added to the information on the Web using RDF as a data model as explained above. However, the form and the meaning of this markup or metadata, specific to each domain, is represented through an ontology. The term *Ontology* was defined by Tom Gruber in 1992 as *a formal, explicit specification of a shared conceptualisation*[Gru93]. In Computer Science, an ontology is seen as a conceptual model that formally and explicitly defines common and shared vocabulary (concepts and relationships between them) across a domain. For example, an ontology in the music domain would contain concepts such as "Album", "Artist", "Writer", and "Performance". These concepts would have properties and relations connecting them such as "name", "releaseDate", "image", "hasArtist", and "performedIn". Ontologies also capture hierarchies found in a domain. Referring back to the music domain example, the ontology would show that the concepts "Artist" and "Writer" are subclasses of the concept "Person" and "Album" is a subclass of the concept "MusicalWork".

Ontologies can either describe a single domain (e.g. Geography), in which they are known as *domain/domain-specific ontologies*, or multiple heterogeneous domains (e.g. Geography, Music and Science), in which they are known as *upper/generic ontologies*. Examples of domain ontologies are *Geonames*[14], *SWDF*[15] and *The Gene Ontology (GO)*[16], covering information from the Geography, Academia and Biology domains, respectively. On the other hand, some of the upper ontologies used in the Semantic Web are *OpenCyc*[17], *SUMO*[18], *DOLCE*[19] and *WordNet*[20].

Using ontologies, assumptions are made explicit and people and machines can share the same understanding of information about a domain. Ontologies also allow the reuse of information found within a domain, as well as help in reasoning over it. RDF-Schema (RDF-S) and OWL were created as extensions for RDF to allow modelling this domain-specific information (classes and relations between them).

2.3.1 RDF-S

RDF-S provides a language that goes beyond RDF in the ability of formally describing the meaning of terminology specific to every domain. That is, to define resources and their types (classes), properties specific to certain types as well as relations that can be found between specific types. RDF-S uses the same syntax as RDF, with extensions added to define this domain-specific information. A class in RDF-S is described

[14] http://www.geonames.org/ontology/documentation.html
[15] http://data.semanticweb.org/ns/swc/ontology
[16] http://www.geneontology.org/
[17] http://www.opencyc.org/
[18] http://www.ontologyportal.org/
[19] http://www.loa.istc.cnr.it/DOLCE.html
[20] http://wordnet.princeton.edu/

by `rdfs:Class`, while a property is described by `rdfs:Property`. The resources of a certain class are known as *instances* of this class and described by `rdf:type`. Additionally, `rdfs:subClassOf` is used to describe the relation between a class and its parent class. Similarly, `rdfs:subProperty` is used to describe the relation between a property and its parent property. Finally, in order to specify how properties can be used to relate certain classes (object properties) or provide literal values for instances of a class (datatype properties), each property in an ontology has a domain and a range. The domain specifies the types of classes that can be found in the *subject* field with this property/predicate in a statement and is described by `rdfs:domain`. The range, on the other hand, specifies the type of classes that can be found in the *object* field, and is described by `rdfs:range`.

The following example illustrates the usage of the above notions to describe domain-specific information. It defines a *Car* class, a child class of *Vehicle* class and also defines a *Door* class. The `hasDoor` is an example of an object property relating the two classes: *Vehicle* and *Door*. Finally, *numberOfDoors* is an example of a datatype property for the *Vehicle* class.

```
<rdfs:Class rdf:ID="Vehicle"/>
<rdfs:Class rdf:ID="Car">
   <rdfs:subClassOf rdf:resource="#Vehicle"/>
</rdfs:Class>
<rdfs:Class rdf:ID="Door"/>

<rdf:Property rdf:ID=hasDoor">
   <rdfs:domain rdf:resource="#Vehicle"/>
   <rdfs:range rdf:resource="#Door"/>
</rdf:Property>

<rdf:Property rdf:ID="numberOfDoors">
  <rdfs:domain rdf:resource="#Vehicle"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</rdf:Property>
```

2.3.2 Web Ontology Language (OWL)

Although RDF-S can be used to describe information within a domain, it is limited in specifying certain logical relations that are required to leverage the potential of reasoning over this information. OWL, the W3C recommendation for publishing ontologies in the Semantic Web, was thus created from this need. Some of the main differences between OWL and RDFS is the ability of the first to define *equivalence* and *disjointness* between classes and to impose *cardinalities* on properties. Two classes are defined as equivalent using the term `owl:equivalentClass`, which is used to relate two classes that share

the same description and that have the same set instances. Another important logical relation is the disjointness, which is defined by `owl:disjointWith` and states that the two related classes can not have any instances in common (e.g. Man and Woman). Furthermore, `owl:cardinality` is used to specify a restriction on the number of values a property can have, within a specific class description.

A very important and extremely useful property defined in OWL is `owl:sameAs`, which is used to state that two individuals/resources refer to the same entity in the real world. This property has been widely used in linking ontologies (diverse sources of information) together, one of the major goals in the vision of the Semantic Web. Another property, similarly used to link identical individuals, is `owl:InverseFunctionalProperty`. The W3C OWL specification states that "If a property is declared to be inverse-functional, then the object of a property statement uniquely determines the subject (some individual). More formally, if P is an owl:InverseFunctionalProperty, then this asserts that a value y can only be the value of P for a single instance x"[21].

The additions described above provide immense opportunities for applications in reasoning over the data, inferring new facts and aggregating pieces of information from different data sources by following links connecting them.

2.4 SPARQL

SPARQL is a W3C recommendation that stands for SPARQL Protocol and RDF Query Language. The main part of a SPARQL query is the *basic graph pattern (BGP)*, which is like an RDF triple except that each of the subject, predicate and object can be a variable. A BGP matches a subgraph of the RDF data when RDF terms from that subgraph may be substituted for the variables, and the result is an RDF graph equivalent to the subgraph. The result of a query is a solution sequence that consists of one or more solution mappings. A solution mapping is a mapping from a set of variables given in the query to a set of RDF terms [PS08]. In addition to BGPs, a SPARQL query can also have one or more solution modifiers such as `LIMIT` and `DISTINCT`, pattern matching constructs such as `OPTIONAL` and `UNION` as well as `FILTERs` for restricting the solution space. An example of a SPARQL query which retrieves the value for the "capital of Spain" is shown below:

```
SELECT DISTINCT ?capital
WHERE
{
      ?uri a <http://dbpedia.org/ontology/Country>.
      ?uri <http://dbpedia.org/property/capital> ?capital.
      ?uri rdfs:label ?label.
      FILTER (regex(?label,"^Spain")).
}
```

A `SELECT` query, such as the above, is the most commonly used form of SPARQL queries. Similar to an SQL SELECT statement, it returns the values of the queried

[21] http://www.w3.org/TR/owl-ref/#InverseFunctionalProperty-def

variables according to the specified constraints. There are three other forms of SPARQL queries which are used for different purposes: `ASK`, `CONSTRUCT` and `DESCRIBE`. The same solution mapping process is performed in an `ASK` query, however the result is returned as "True" or "False" stating whether there is an answer for the query or not. `CONSTRUCT` is similar to both forms in the query structure with the result being an RDF graph representing the solution sequence. Finally, `DESCRIBE` is rather different than the previous three forms. It is used to retrieve an RDF graph containing information about the queried resources, with the exact description of the information being determined by the query service/processor.

2.5 Linked Data

The Web currently comprises a Web of documents that people can read or follow links from to other documents, while the Semantic Web is a Web of data that is interlinked through relations expressed in RDF. That is where Linked Data fits in the context of Semantic Web. The Semantic Web, or Web of Data, is the goal or the end result, while Linked Data provides the means to reach that goal [BHBL09].

> "The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web." [BHBL09]

There are four principles for publishing Linked Data, which were outlined by Tim Berners-Lee in his article [BL06] as follows:

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

Because URIs uniquely identify resources and things on the Semantic Web, they should be carefully chosen. [BCH07] explains a list of best practices for choosing a URI, some of which are as follows:

1. Use HTTP URIs for everything.
2. Define URIs in an HTTP namespace under your control where they can be dereferenceable.
3. Keep URIs away from implementation details. Consider the difference between the following examples:
   - http://dbpedia.org/resource/Berlin
   - http://www4.wiwiss.fu-berlin.de/dbpedia/cgi- bin/resources.php?id=Berlin
4. Keep URIs stable and persistent as changing them will break any already-established links.

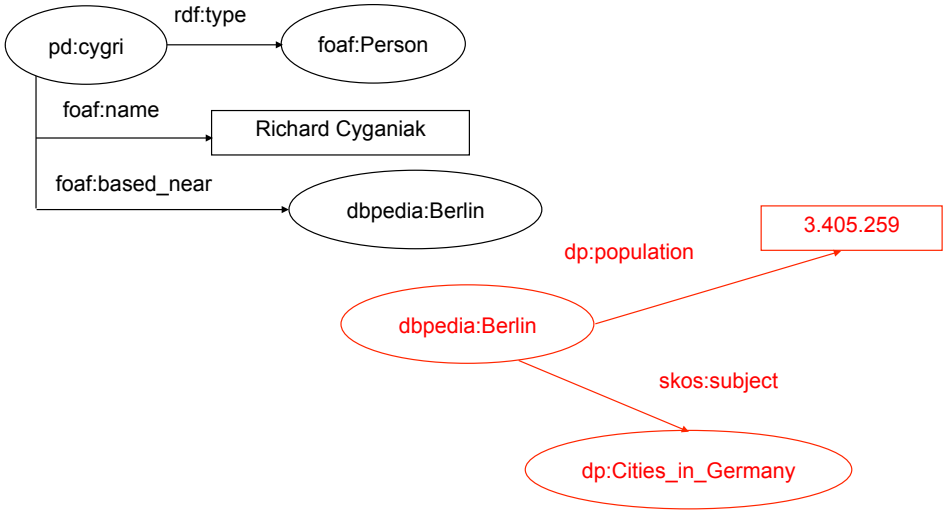


Figure 2.2: Linking datasets using URIs

Connecting different datasets on the Web is as important as following the standards of publishing them. Figure 2.2 [22] shows one way of linking data that is through the reuse of URIs. The figure uses concepts, instances and relations found in well known datasets such as foaf[23] and dbpedia[24]. FOAF (Friend of a friend) represents an ontology for describing people, relations between them and things they create and activities they do. DBpedia is a linked data set created as a result of a community effort to extract structured information from Wikipedia. The DBpedia knowledge base currently describes more than four million things. As both datasets in the figure are using the URI *http://dbpedia.org/resource/Berlin* to represent the city Berlin, they get naturally and implicitly connected.

[22] URIs are omitted for simplicity of the graph
(dbpedia:Berlin = `http://dbpedia.org/resource/Berlin`)

[23] `http://www.foaf-project.org/`

[24] `http://dbpedia.org/About`

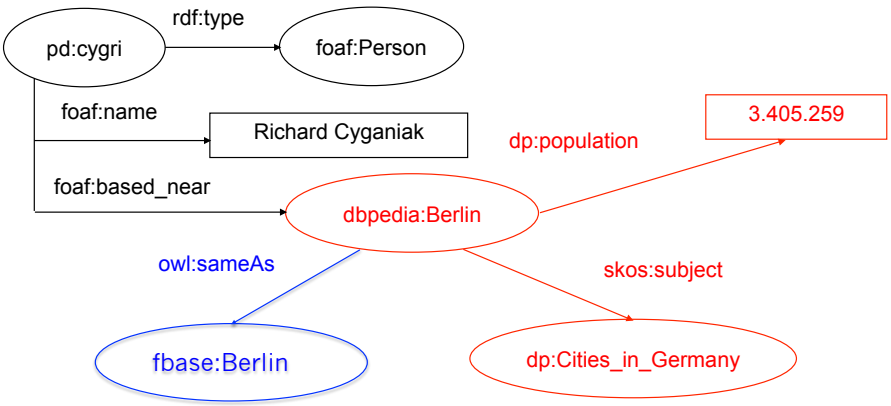


Figure 2.3: Linking datasets through relations

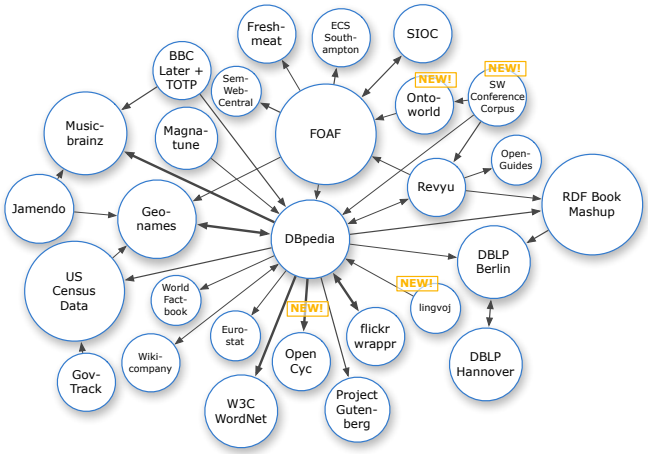


Figure 2.4: Linked Data as of November 2007

Another way of linking different sources of data is by explicitly establishing relations between them. Figure 2.3 shows how this is achieved by connecting the dbpedia and freebase[25] datasets. Freebase is an open data set covering categories and data from other large datasets like Wikipedia, MusicBrainz, and the SEC archives. The contained data spans topics such as movies, people, locations and music. The example in the figure uses the relation *owl:sameAs*[26] which gives the ability to express equivalences between seemingly different individuals. Thus, connecting *dbpedia:Berlin* and *fbase:Berlin* using *owl:sameAs* states that they are actually the same thing.

By publishing and linking different data sources on the Web, people are actually helping in realising the vision of the Semantic Web. Although the term Linked Data was coined by Tim Berners-Lee, it was not until late 2006 when he published an article about Linked Data and its rules and principles [BL06] that things began to change and people started to put data on the Web. DBpedia was the first dataset exposed on the Web. Figure 2.4 presents the datasets that have been published in Linked Data format as of November 2007, which shows that steps were still slow.

Early in 2009, in his talk at TED[27], Tim Berners-Lee asked people to put raw data up on the web, to start publishing linked data and to connect different datasets together. Figure 2.5 shows the progress of this as of September 2011. The big difference in the two figures indicates a progress towards realising the Web of Data.

2.6 Summary

This chapter has provided an introduction to the Semantic Web and various concepts and technologies which are used within the community and throughout the thesis. These include *RDF* as the standardised data model for representing information in a machine-

[25] http://www.freebase.com/
[26] http://www.w3.org/TR/owl-ref/#sameAs-def
[27] http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html

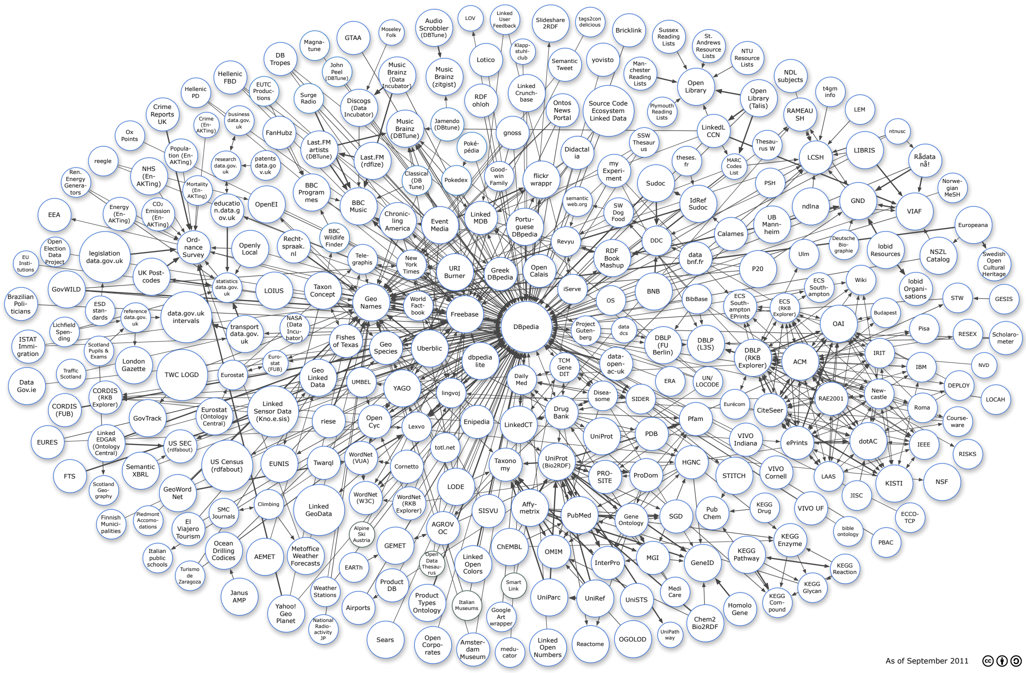


Figure 2.5: Linked Data as of September 2011

processable format; *RDF-S* and *OWL*, as ontology languages, which go beyond RDF in defining common vocabulary specific to each domain; and *SPARQL* as the main query language for this semantic data. These technologies and languages have enabled a huge amount of information to be expressed in a structured format, allowing a wide range of tasks and applications using this data such as sharing, reusing, reasoning over and aggregating different pieces of this information. The notion of *Linked Data* has also helped in realising the vision of the Semantic Web and has turned it into a global database containing information about all types of real-world entities. All this revealed new opportunities for changing the way search is done, as will be discussed in the next chapter which provides a thorough review of literature of semantic search.

Chapter 3

Semantic Search

3.1 Introduction

Search has evolved from being a task performed by trained librarians to a fundamental every-day task performed by non-expert users. At present, information retrieval (IR) systems provide these users the means to access large, heterogenous distributed archives of information. The traditional IR paradigm – also referred to as keywords-based or syntax-based – adopted by some of these systems considers users' queries as bags of words for which it attempts to find the best matches in an index. Since retrieval is usually based on syntactic matching between the query and the indexed documents, this could harm both precision (since a single word could have more than one meaning which causes false matches) and recall (since multiple words could have the same meaning which causes true matches to be missed) of the results.

Several authors have proposed to overcome this limitation by attaching semantics to the data. This is the vision of the Semantic Web: to create an environment in which the meaning of information is explicit for processing by machines. Additionally, the Semantic Web is about publishing data (e.g. people and places are resources that have URIs) and connecting it, which would turn the Web into a global database that can be queried to find answers for various information needs. Together, these provide immense opportunities for addressing the above limitations and changing the way search is done. For instance, understanding the semantics of the data and reasoning on it, and integrating pieces of information from different data sources by following links connecting them are some of the opportunities offered by the Semantic Web for so-called 'semantic search'.

This chapter introduces the concept *semantic search* and reviews the literature in this area. The rest of the chapter is structured as follows: Section 3.2 defines terminology and types of search to be addressed throughout the thesis. Section 3.3 describes the different formats employed for formulating the input query and reviews examples of semantic search systems within each format. Section 3.4 discusses the amount of transformation required for the user's query, in each format, before execution. Section 3.5 describes the main steps required to execute the translated query against the

search space describing one (closed-domain) or more domains (open-domain). Finally, Section 3.6 discusses the different formats adopted by the reviewed semantic search systems for results presentation.

3.2 Defining Semantic Search

No unified definition of *semantic search* exists. It has been used by different research communities including Information Retrieval, Natural Language Processing and the Semantic Web to describe different approaches and strategies employed to improve search performance and user experience. However, they all share the broad goal, which is to better understand users' information needs (represented in their queries) and/or the Web/domain content; and to improve the matching required between performance and experience. The following list summarises the most common ways in which the term 'semantic search' has been used within these communities:

A. Using query expansion (e.g. using synonyms) and/or natural language techniques to better understand the user query and in turn improve retrieval performance (of unstructured data).

B. Using statistical smart indexing as well as other information retrieval techniques to better understand the unstructured indexed data and in turn improve retrieval performance.

C. Using Semantic Web data (e.g. RDF documents, RDFS/OWL ontologies and RDF datasets including linked data) to enrich search results (returned from searching traditional web pages such as HTML).

D. Searching Semantic Web data (e.g. RDF documents, RDFS/OWL ontologies and RDF datasets including linked data) and returning answers as a list of links to these resources (e.g. links to documents or ontologies).

E. Searching Semantic Web data (e.g. RDF documents, RDFS/OWL ontologies and RDF datasets including linked data) and returning answers resulting from reasoning on the data found in these resources.

In this thesis, the discussion is restricted to the approaches covered by the definitions given in points D and E. Figure 3.1 shows an abstract architecture for semantic search in which the basic steps in the search process are illustrated. The user inputs their query in a specific input format that is adopted by the system (e.g. as a NL sentence or using a view-based interface to construct the query). The different query input approaches are discussed in Section 3.3. The query is then processed and transformed into a formal representation as required by the underlying query engine. The amount of transformation is influenced by the query input approach as shown in Figure 3.1 and discussed in Section 3.4. The formal query is then executed against the search space which either describes a single domain (closed-domain) or multiple ones (open-domain). This step is discussed in Section 3.5. Finally, results generated from this

step – documents or data – are presented to the user in a format chosen by the system (e.g. ranked list of documents or NL answers). Results integration and presentation is discussed in Section 3.6.

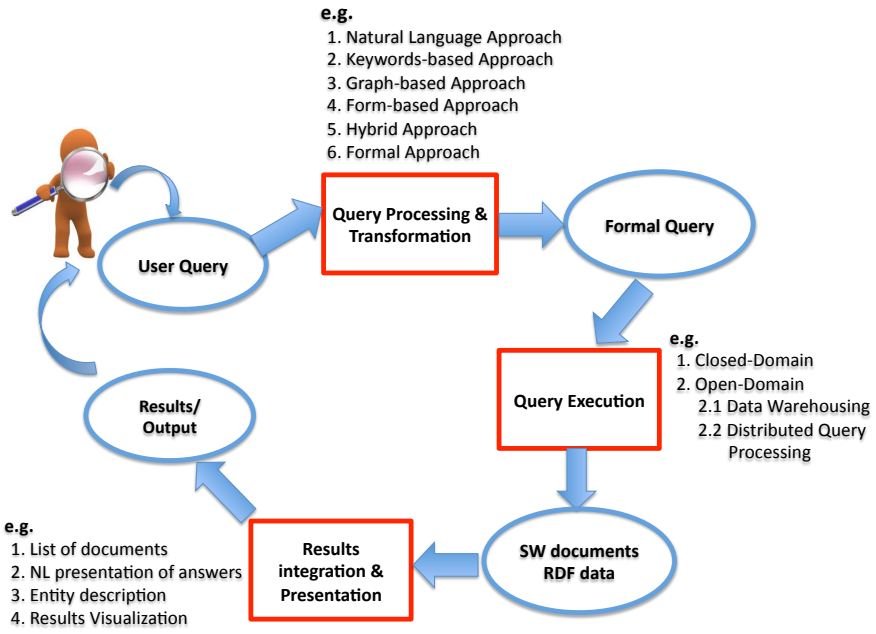


Figure 3.1: Abstract architecture for semantic search

The strategy followed to identify relevant work to include in this review is as follows: first, a wide range of publications including journal articles, conference proceedings, workshop proceedings, theses as well as chapters from books have been collected. Journals referenced include: Semantic Web Journal, Journal of Web Semantics, Knowledge Engineering Review, SIGIR Forum, International Journal of Human Computer Interaction, Information Processing and Management, and International Journal on Semantic Web and Information Systems. Conferences referenced include: International Semantic Web Conference (ISWC), World Wide Web Conference (WWW), Extended Semantic Web Conference (ESWC), International Conference on Knowledge Engineering and Knowledge Management (EKAW) and International Conference on Knowledge capture (K-CAP). Major workshops addressing this field are also referenced including: Semantic Search (SemSearch2010), Question Answering Over Linked Data (QALD2011), Interacting with Linked Data (ILD2012), Linked Data on the Web (LDOW2009), Consuming Linked Data (COLD2010) and Evaluation of Ontologies and Ontology-based tools (EON2007). These publications allowed identifying the most important challenges facing semantic search systems (which fall under the restricted definitions described above) in each step of the search process (shown in Figure 3.1). The different approaches and techniques presented in these publications are then reviewed and organised with respect to the common methodologies followed in tackling the identified challenges.

3.3 Input Query Format

The interaction between user and system is vital to the success of any search. Semantic search operates over structured data, which is harder for users to comprehend than textual data. The input interaction therefore needs to support users to comprehend what they may sensibly ask. In order to make use of the opportunities offered by the Semantic Web, it is important to identify the best interaction paradigms or query interfaces/formats. Users' experience and satisfaction with the information seeking process is, indeed, influenced by many other aspects including the performance of the search system (in terms of retrieval and responsiveness) as well as the presentation of the results returned, but the query format is the starting point. It is the place at which users can be guided to make their query in a way that will produce relevant results. The main difference which affects the kinds of results which may be obtained is the expressiveness of the query language adopted, but the usability of the interface and the kinds of support provided during query formulation can in practice make a great deal of difference to whether users can successfully express their queries. Thus, the main challenge for semantic search approaches in the input query formulation is to identify and adopt the query format that provides the highest (balanced) level of expressiveness and ease of use. In the rest of this section, we review the different query formats adopted in the literature of semantic search with respect to these aspects. These formats tend to fall into one of the following categories:

- *Formal (Structural) approach*: The input query is expressed in one of the formal query languages for RDF (e.g. SPARQL or SeRQL) which are used to retrieve data from an RDF model. For example, SQUIN [HBF09] takes a SPARQL query as input.
- *Natural Language (NL) approach*: The input query is expressed using a natural language such as English (e.g. 'Where is the University of Sheffield located?'). For instance, PowerAqua [LMU06] and FREyA [DAC10] accept free-form queries including keywords, phrases or full questions.
- *Keywords-based approach*: The input query is a set of keywords of interest to the user (e.g. 'location University Sheffield'). Some of the systems employing this approach are Swoogle [DFJ+04], Watson [dBG+07] and Sindice [TOD07].
- *Graph-based approach*: The input query is formulated using a graph-based interface that explores the search space. Semantic Crystal [BKGK05] employs this approach to aid users in constructing their queries by visualising the data available and the possible ways of querying it.
- *Form-based approach*: This approach is similar to the graph-based approach in visualising the search space, while being different in using forms instead of graphs as the interface to build the query. Corese [CDKFZG06] uses forms that have check boxes and drop-down lists, which allow the user to specify the properties required for a parameterized search.
- *Hybrid approach*: This approach uses a combination of the previous approaches as the query format. [BCC+08] used keywords, in addition to forms, in implementing

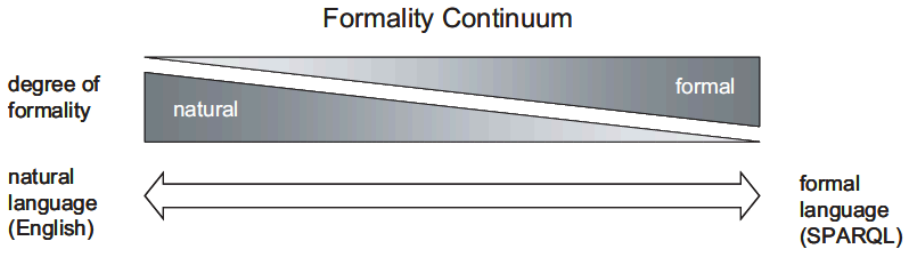


Figure 3.2: The Formality Continuum [Kau07]

their system K-Search. [HV03] integrated view-based[1] navigation of the underlying repository with keywords-based search.

In the following sections, we discuss in more detail exemplar systems for each approach. Table 3.2 provides a summary of systems and approaches.

[Kau07] placed the formal and natural language approaches at the ends of a Formality Continuum as shown in Figure 3.2. The degree of formality influences the query language's expressiveness and usability. Both aspects test the usefulness of the query language in helping users express their information needs and formulate searches [ULL+07]. The expressive power of a query language specifies what queries a user is able to pose [AG08].

3.3.1 Formal Approach

Semantic search systems adopting a formal query input approach use one of the RDF-based formal languages to query the RDF model (e.g. SPARQL, SeRQL). In terms of usability and usefulness, this approach requires users to learn the underlying query language. This can be acceptable for developers of Semantic Web applications and experts but not for non-expert users. A non-expert user would feel most uncomfortable trying to learn a formal query language to answer their information needs [Kau07]. On the other hand, with respect to expressiveness, more complex queries can be formulated using this approach since the same structure found in the data (e.g. relations between concepts) is also applied in the queries. Formal query languages differ in their expressivity compared to each other. For instance, as stated by [BBFS05, p.54], the *RQL family* consisting of the language RQL [KMA+03] and its extensions such as SeRQL [BK03] is far more expressive than the *SPARQL family* which originated with SquishQL [MSSR02], evolved into RDQL [MSSR02] and then later extended to SPARQL [MSSR02].

SQUIN is an example in this category that accepts SPARQL queries as input to find answers in the Web of Data [HBF09]. An example of a complex query that can be answered in SQUIN is 'Find all developers of the Tabulator Project, their email addresses and other projects they are involved in'[2]. The equivalent SPARQL query is

[1] [ULL+07] define a separate category for View-based systems as those using ontology presentation and ontology navigation to support query construction and domain exploration. We see this as describing both form-based and graph-based approaches in our classification.

[2] taken from http://www4.wiwiss.fu-berlin.de/bizer/ng4j/semwebclient/

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX doap: <http://usefulinc.com/ns/doap#>
SELECT DISTINCT ?name ?mbox ?projectName
WHERE {
  <http://dig.csail.mit.edu/2005/ajar/ajaw/data#Tabulator> doap:developer ?dev .
  ?dev foaf:name ?name .
  OPTIONAL { ?dev foaf:mbox ?mbox }
  OPTIONAL { ?dev doap:project ?proj .
             ?proj foaf:name ?projectName }
}
```

Figure 3.3: Example of a SPARQL query as input to SQUIN

shown in Figure 3.3.

3.3.2 Keywords-based Approach

In this approach, the input query is given as a set of keywords that represent the user's information need. This is the classical approach used by most traditional search engines such as Google and Yahoo! as well as Semantic Web search engines such as Sindice [TOD07] and SWSE [HHD+07].

Usability studies in the literature, which assess the usefulness of this approach, showed contradictory results. While [Kau07] found that users preferred using natural language queries to keywords, [RLME05] showed that students generally preferred keyword-based search over full-questions search. Positive comments given by users in the former study included 'easy to use; no thinking required; robust to input' while negative comments included 'query language not clear'. The authors in the latter explained their findings by suggesting that students could be more accustomed to use keyword search engines, which agrees with [TCRS07]. Further, they concluded that entering keywords is an easier and more comfortable task than entering whole sentences.

Although [DN09] argue that keywords-based approach suffers from limited expressivity when compared to other approaches, [LUM06] claim that their semantic search system *SemSearch*, which accepts keywords as input, allows end users to ask complex queries (compared to simple keyword search). For example, the query 'Give me the cities of the Universities in England' would be formulated in SemSearch as '`cities: universities England`' where ':' is used to explicitly specify the subject that is the expected type of the search results. Moreover, the authors claim that their approach provides a more flexible way of specifying queries than the form-based approach.

3.3.3 Natural Language (NL) Approach

In this approach, the input query is expressed using a natural language such as English or French. The exact form of the query, as well as how much freedom the user has in formulating the query, vary among different systems. For instance, while some systems accept free-form queries such as phrases or full sentences, others might require a specific query format (e.g. only full sentences) or accept only certain questions (e.g. WH questions – such as "What" or "Where"). Querix [KBZ06] requires full English questions

with restriction to the sentence beginnings (sentence must start with 'Which', 'What', 'How many', 'How much', 'Give me', or 'Does'). Querix was chosen by the users in the [Kau07] study as the system with the the most preferred query language. Although it can be seen to have a restricted language (full, grammatically-correct sentences with specific beginnings), users found it completely free and natural.

[Kau07] showed in their usability study that the NL approach was judged by users to be the most useful and preferable. The authors attributed this finding to the fact that users can communicate their information needs in a familiar and natural way without having to think of appropriate keywords. The same study found that people can express more semantics when they use full sentences rather than keywords. Similarly, [DN09] state that NL queries offer users more expressivity to describe their information needs than keywords.

However, the NL approach suffers from both syntactic (related to the structure of the sentence) as well as semantic (related to the meaning of the words) ambiguities. The performance of the NL processing techniques used for parsing and analysing the sentences influences the systems employing this approach. Because of this, the development of those systems is usually very complex and time-consuming [Kau07].

Another limitation faced by the NL approach and similarly by the keyword-based approach is the lack of knowledge of the underlying search space by the users. The result is that users input their own query terms which are usually different from the ones expected by the system. This is an acknowledged problem in literature, as stated by Kaufmann et al.: "*However, the interface ... is inherently affected by the habitability problem due to its flexible natural query language*"[KB10, p.2]. The outcomes of the usability study presented in Chapter 7 showed how this problem not only affects the performance of the NL system but also the user satisfaction. The NL-system evaluated in the study (NLP-Reduce) got the lowest success rate (20%) together with the highest number of attempts (4.1 on average and 8 as a maximum) performed to answer a specific query. The latter is due to the users having to rephrase their queries to substitute the words the system is expecting. This was also supported by the most repeated negative comment given by the users of this system: 'I have to guess the right words'. The users found that they could get answers with specific words rather than with others. For instance, using 'run through' after river returns answers which are not given when using 'traverse', or accepting abbreviations in some queries and not in others. This problem is the main challenge facing natural language interfaces [LUSM11, KB10, ULL+07].

In an attempt to overcome this problem, some systems employ a controlled NL approach which only accepts query terms that are valid/found in the system's own vocabulary. For instance, [BKK05] follow this approach: Ginseng offers suggestions to the user according to a specific grammar and refuses entries that are not in the possible list of choices. Again, the same usability study – mentioned above – showed that while the guidance through suggestions of valid terms and the prevention of invalid ones offered the most support and confidence for non-expert users, it was perceived by the expert users to be very restrictive rather than helpful. This was due to the system not allowing users to input their own query terms, which was frustrating, particularly

when they got stuck and did not know how to continue. Recall how we earlier discussed the expressiveness of the query language and its effect on the users' ability to formulate their information needs and on their overall satisfaction. This is supported by the results of the same study showing how the limited expressivity of Ginseng caused expert users to have an unsatisfying experience (indicated by the fact that it was the least liked interface).

Table 3.1: Suggestions generated by FREyA for the property 'population' in the query 'Which city has the largest population in California?' to support the user in formulating superlatives and comparatives [DAC10]

Query	**Suggestions**
Which city has the largest population in California?	1. Max (city population) 2. Min (city population) 3. Sum (city population) 4. None

Referring back to expressiveness, a challenge that needs to be addressed by all query approaches is the support for superlatives and comparatives in queries. One way to face this challenge is to engage the user while attempting to understand the query. For instance, the approach adopted in FREyA [DAC10] is to ask the user to identify the correct choice from a list of suggestions whenever a numeric datatype property is identified in the query. To illustrate, consider the query 'Which city has the largest population in California?'. As a result of identifying 'population' as a numeric datatype property, FREyA generates maximum, minimum and sum functions. The generated suggestions for the query example are shown in Table 3.1. The user can then choose the correct superlative or comparative depending on their needs.

In the same context, another approach has emerged recently that provides increased expressiveness based on using (predefined or generated on-the-fly) templates to capture the semantic structure of NL queries (adopted in TBSL [UBL+12] and QAKiS [CAC+12]). This allowed TBSL to support more complex queries, such as those containing quantifiers, comparatives or superlatives. An example of such queries is: "*How many films did Leonardo DiCaprio star in?*" for which TBSL generates the following template:

```
SELECT COUNT(?y) WHERE {
  ?y rdf:type ?c.
  ?x ?p ?y.
}
Slots:
 < ?x, resource, Leonardo DiCaprioi >
 < ?c, class, films >
 < ?p, property, star>
```

To generate these templates, TBSL makes use of Pythia's parsing capabilities [UC11], which depend on a dictionary to produce both syntactic as well as semantic representations for an expression using a Lexicalized Tree Adjoining Grammar [Sch90] and representations similar to Underspecified Discourse Representation Theory. Part of the dictionary was manually created to cover generic, ontology-independent terms such as *'most', 'least', 'give me'* or *'have'*. The ontology-dependent part is, on the other hand, generated while parsing each query. A POS tagger and a set of heuristics are used for this task. For instance, nouns are mapped to both classes and properties, verbs are mapped to properties, and noun phrases are mapped to instances. The template would thus contain empty slots (as shown above) for each ontology-dependent component to be later filled with its URI.

Although this deep linguistic and semantic analysis allow both TBSL and Pythia to support more complex queries, this approach has several limitations: 1) relying on the manually-created lexicon prevents it from scaling to very large datasets, 2) relying on fixed templates does not guarantee a suitable match for all types of questions and finally, 3) this approach relies heavily on the structure of the NL question given by the user, which is not guaranteed to be a complete and grammatically-correct question.

3.3.4 Form-based Approach

Systems employing forms for query input attempt to support users in constructing their queries by visualising the search space. Additionally, this approach benefits from overcoming the habitability problem, earlier discussed, that faces both keywords-based and NL-based approaches. This is implicitly achieved since the users build their queries by selecting terms (concepts, relations or instances) shown in the interface. Corese [CDKFZG06] uses a form-based interface for users to query a specific domain. The forms have check boxes and drop down lists that allow users to specify the properties required for a parametrized search. Corese received very positive comments from its users including appreciation for its form-based interface.

Additionally, KIM [KPT+04], one of the earliest systems applying semantic search, adopted a form-based query approach for documents' annotation and semantically-enhanced information retrieval. The system was intended to help perform automatic annotation of documents through extraction of classes and entities and mapping them to ones found in the underlying knowledge bases. Then based on this annotation, KIM provided its users with the ability to retrieve documents referring to these ontological terms.

Explanations for whether to adopt or avoid the form-based approach have been contradictory in the literature. On one hand, forms can be helpful to explore the data in the search space and understand the possible ways of querying it [ULL+07]. Additionally, the usability study I conducted – presented in Chapter 7 – showed that users found the form-based approach less complex than the graph-based approach while providing them with the ability to perform more complex queries than with the NL approach. On the other hand, the exploration of the search space can be a burden on users that requires

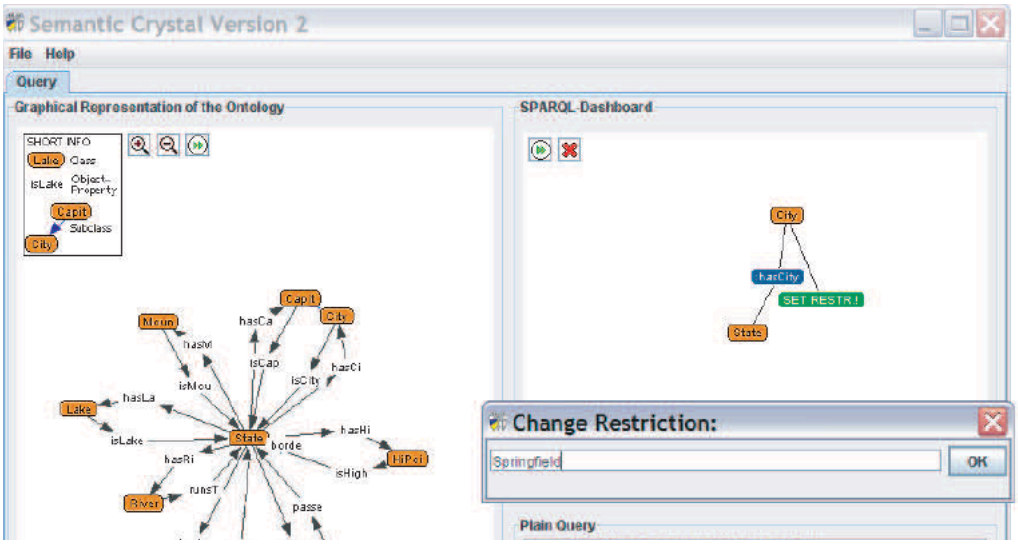


Figure 3.4: With datatype properties, a user can specify a restricting value such as 'Springfield' [Kau07].

them to be familiar with the underlying ontology and semantic data [LUM06]. Furthermore, forms can become tedious to use, especially in large spaces [LMUS07]. This is because of technical limitations such as the number of items that can be included in a scrolling list or a tree-like view [ULL+07] as well as the convenience of users in visualising, inspecting and locating the required terms (e.g. concepts and relations between them). Finally, it seems impossible to adopt this approach in heterogenous spaces like the open web, as it is not clear what would be visualised in this case. This clearly limits the usability of the form-based approach in an open-domain environment.

3.3.5 Graph-based Approach

Graph-based systems employ graphs, rather than forms, for the same purposes described above: visualising the search space, supporting query formulation and side-stepping the habitability problem. To illustrate, the user interface of Semantic Crystal is shown in Figure 3.4.

The graph-based approach shares most of the advantages, as well as the limitations, of the form-based approach. One of the differences, however, is that graphs show the structure of the data with the concepts and the relationships between them clearly plotted, which can provide users with a direct understanding of the search space and the possible valid queries. However, being able to explore large or complex search spaces is more challenging for systems adopting this approach [ULL+07]. They face the same technical limitations as form-based approaches on how much can be presented to the user. Moreover, they can get more cluttered, which affects the usability of the interface and in turn the ability of users to efficiently and effectively formulate searches. Figure 3.4 highlights this problem: the visualised ontology contains only eight concepts which makes it small compared to others – commonly found in the Web of Data – such

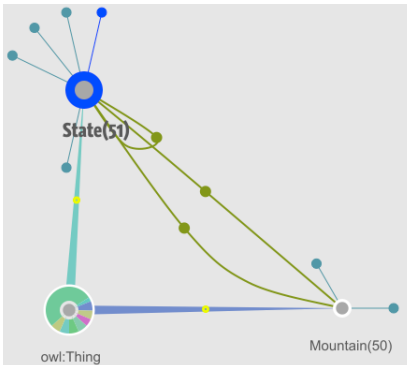


Figure 3.5: Affective Graphs showing only concepts selected by a user.

as DBpedia. In the first case, the graph is clear and can be easily explored, however, as the ontology gets bigger, the screen can easily get scattered with concepts and arrows representing relations between them.

In an attempt to tackle this challenge, Affective Graphs [3] opts for expanding only the concepts and relations which get selected by the user, rather than the whole ontology. The interface adopted by Affective Graphs is shown in Figure 3.5. Users are presented with a force directed graph, where nodes represent concepts and links represent properties or relations. Links connecting two concepts can represent subclass relations or object properties while unconnected links represent datatype properties. These unconnected links are visually represented as straight lines arising out of a node. As shown in the figure, only concepts selected by the user, for example, State and Mountain, are visualised, together with their properties.

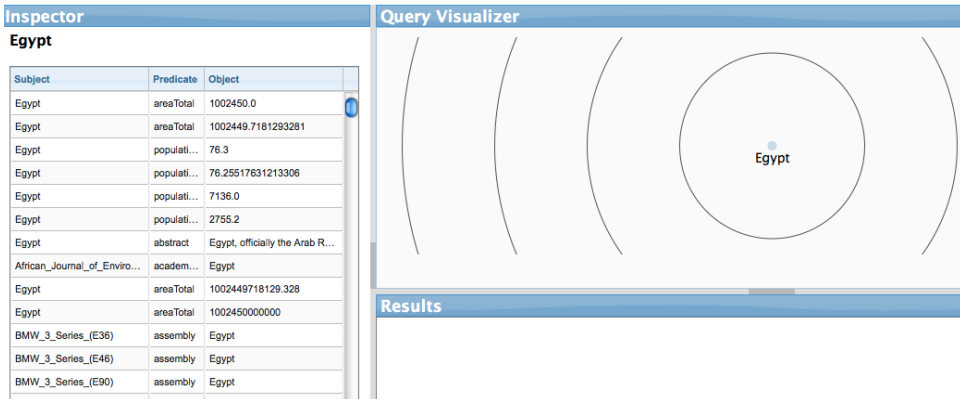


Figure 3.6: Results returned by Smeagol for the query term '`Egypt`'. As explained by [CD11], "the query visualizer pane (top-right) displays the user's current subgraph. The subgraph is depicted using a radial layout algorithm. The advantage is one of locality: the resource in the center of the visualization is the one currently most relevant to the user; it is also the resource shown in the inspector pane".

[3] `http://oak.dcs.shef.ac.uk/?q=node/253`

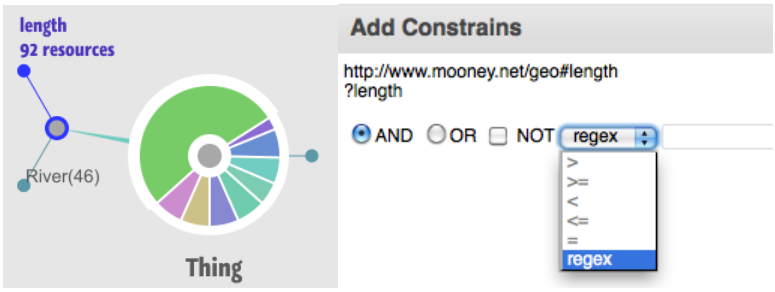


Figure 3.7: Affective Graphs support for comparatives with numeric properties.

In a similar way, and to identify a specific area of interest, Smeagol [CD11] introduces a "specific-to-general" graph-based interface where it starts from an entity or a term entered by the user (e.g. 'Egypt') and builds a related subgraph extracted from the underlying data. After the user disambiguates the query term from a list of candidates, the system returns a list of triples containing that term for the user to select from and add to their specific subgraph of data. However, in a dataset such as DBpedia, this list will often contain thousands of triples for the user to examine in order to select the required ones (see Figure 3.6).

As discussed earlier in Section 3.3.3, a challenge that is only addressed by few approaches is the support for more complex queries such as those containing comparatives or superlatives. In this context, *Affective Graphs* provides its users a way to include comparatives for numeric properties in their queries as shown in Figure 3.7. Whenever a datatype property is selected by the user, the system prompts them to choose from a list of functions that cover comparatives (e.g. 'more than', and 'less than').

Another issue faced by form- and graph- based approaches is the time required to build a query. Building queries by exploring the search space can be time-consuming, especially as the ontology gets larger or the query gets more complex. This was shown by [Kau07] in their usability study in which users spent the most time when working with the graph-based system (Semantic Crystal). This is also supported by the feedback given by the users with respect to this aspect, with comments such as 'too laborious'; 'required many clicks and commands to do a query'; and 'time-consuming'. Additionally, in the usability study presented in Chapter 7, some users mentioned that although being time-consuming, graph-based approaches can be fun and interesting to use and thus could be more suitable for users with specific information needs or certain usage – for instance, infrequent complex queries – as opposed to everyday use.

A query example given by [Kau07] to formulate in Semantic Crystal is 'Which states have a city named Springfield?'. To build this query, a user would first click on the class 'State' which will cause the interface to list the class properties. The following step is to choose the property 'hasCity'. The upper right of the interface shows the user the elements selected at each step, as shown in Figure 3.4. Since 'hasCity' is an object property connecting the classes 'State' and 'City' together, the latter is added to the user query. Then the user can select the datatype property 'name' with the class 'City'

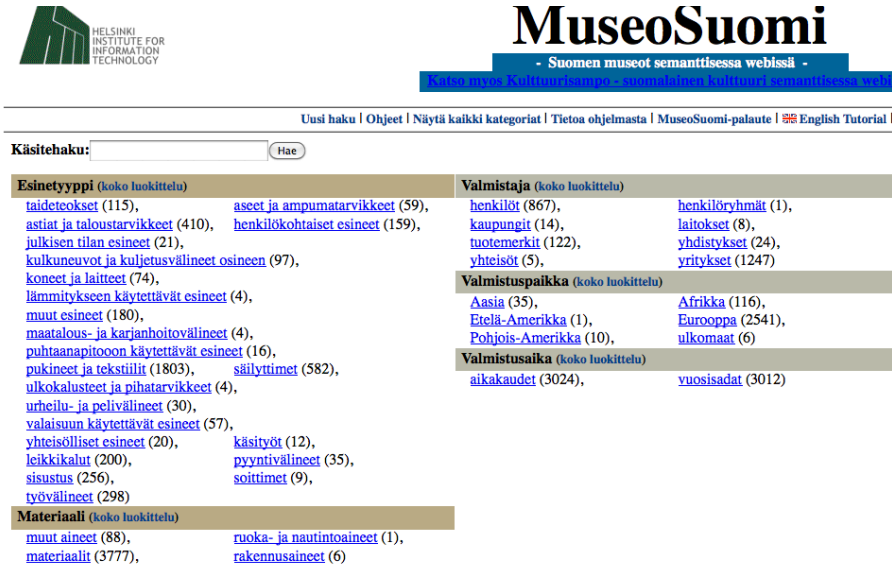


Figure 3.8: User interface of MuseumFinland based on a multi-faceted approach to explore the search space.

to restrict the search to those named 'Springfield'. This brief example shows the steps, effort and time required to build this simple query in a graph-based system.

3.3.6 Hybrid Approach

[HV03] showed that keyword search and view-based search complement each other. Their methodology is based on mapping the underlying domain ontologies into facets, which facilitates multi-facet search[4]. Facets describe general categories such as 'Happenings', 'Persons and roles', and 'Places' which are found in the tourism domain. The facets provide different views into the domain, and aid the users in focusing their information needs and in formulating queries. The multi-facet search is based on a set of hierarchy rules which are themselves a set of configurational rules that tell how to construct the facet hierarchies from the domain ontologies.

The multi-facet approach helps the users most when they do not know what they are searching for, allowing them to explore the search space. However, in order to avoid being a time-consuming task when the users do know what they are looking for, a semantic keyword searching functionality can be used to speed up the query process. [HV03] implemented Ontogator to reflect their methodology which was used in the MuseumFinland system [Mäk06]. They explain how the keywords search functionality is applied as follows: "The search keywords are matched against category names in the facets as well as text fields in the metadata. Then, a new dynamic view is created in the user interface. This view contains all categories whose name or other defined property value matches the keyword. Intuitively, these categories tell the different interpretations of the keyword, and by selecting one of them a semantically disambiguated choice can be

[4] [HV03] use *facets* interchangeably with *views* and *multi-facet* with *view-based*

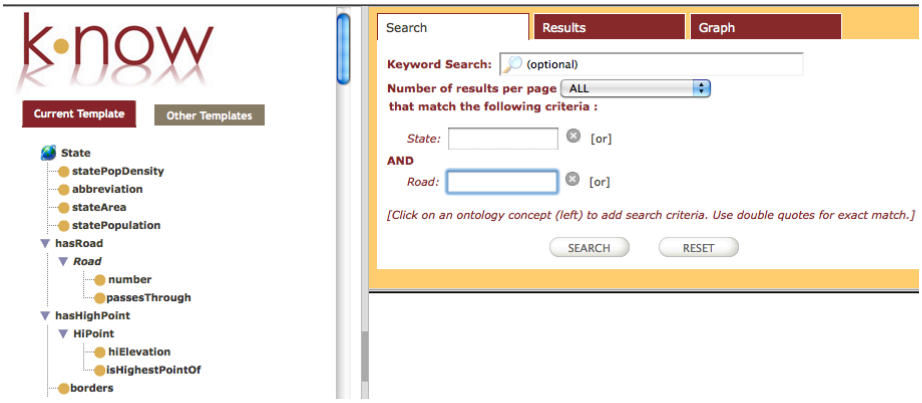


Figure 3.9: User interface of KSearch: forms are used for semantic search and the text field for keywords-based search

made". Figure 3.8 shows the interface of MuseumFinland which relies on Ontogator as the search component.

While [HV03] combined multiple query formats to support users in formulating their search queries and offering them flexibility in expressing their information needs, [BCC+08] combined keywords with forms to implement what they termed "Hybrid Search", a method that was shown by the authors to outperform both keyword-based search and pure semantic search in terms of precision and recall. The authors defined Hybrid Search to be "the application of semantic (metadata-based) search for the parts of the user queries where metadata is available, and the application of keyword-based search for the parts not covered by metadata". This approach was employed in implementing K-Search [BCC+08], which was tested with end-users. The outcomes of the evaluation showed that users especially value the full power of the Hybrid Search concept. Usability studies generally focus on testing the user's ability to form queries and the usability of the query interface as experienced by the users. However, this evaluation aimed at measuring the users' comprehension level of the Hybrid Search paradigm, i.e. the quality of the knowledge retrieval, and the users' judgement of the returned answers' adequacy, i.e. the quality of the document retrieval [BCC+08]. The interface of K-Search is shown in Figure 3.9. Forms are used as the query interface for the semantic search part and the common text field is included for users to input keywords.

3.3.7 Summary

The different query approaches described above offer varying degrees of expressiveness and support during query construction. The formal query approach can be used to build highly complex queries, however, it requires learning a formal query language and, therefore, is not suitable for non-expert users. In contrast, users can express their information needs in a natural language (e.g. English) with both keywords- and NL-based approaches. However, both approaches face a major problem which is the mismatch between the terms found in users' queries and those understood by the semantic search

Table 3.2: Semantic search systems review for user's query format

Systems	**User Query**					
	Formal	Natural Language	Keywords-based	Graph-based	Form-based	Hybrid
Affective Graphs				√		
AquaLog		√				
Corese					√	
DARQ	√					
FalconS			√			
FREyA		√				
Ginseng		√				
[HHK+09]	√					
K-Search					√	√
Librarian		√				
LOQUS	√					
Nlp-Reduce		√				
Ontogator						√
Panto		√				
PowerAqua		√				
Pythia		√				
QAKiS		√				
Querix		√				
Semantic Crystal				√		
SemSearch			√			
Sig.ma			√			
Sindice			√			
Smeagol				√		
SQUIN	√					
Swoogle			√			
SWSE			√			
TBSL		√				
Watson			√			

system (the *habitability problem*). One way to overcome this is to apply a controlled-NL approach which provides guidance through suggestions of valid query terms but while also preventing invalid queries through the use of a restricted vocabulary. Although this approach can provide great support, especially for non-expert users during query formulation, it can be frustrating and can limit users' ability to express their needs. Additionally, the support given to the users in understanding the search space is still limited since neither the structure of the data is shown, nor how it is connected. This is offered by view-based (graph- and form-based) approaches which expose the structure of the ontology to help understand the search space and the possible ways of formulating queries. They attempt to bridge the gap between the users and the system by showing them the data, how it is connected and how it can be linked to construct valid queries. This also allows users to construct more complex queries. However, in terms of usability, they can be difficult to use, especially when used to query large datasets (e.g. the open Web of Data). Also, the graph-based approach can be complicated, especially

for non-expert users. Finally, view-based approaches are the most laborious and require the most amount of time to construct queries.

3.4 Query Processing and Transformation

The amount of transformation done over the user's query before execution depends on the format of the query required as input from the user, the system's degree of domain dependency as well as the underlying query engine. For instance, a system that requires a structured query (e.g. SQUIN[5]) would need no processing as the query is already in a format that can be directly used for execution against the relevant sources. In contrast, a system that accepts a natural language query as an input would need to employ various techniques to transform it into a suitable format for execution.

Similarly, systems searching an open-domain (spanning multiple domains such as Geography, Music and Science) in contrast to a closed-domain (covering a single domain such as Geography)[6] face various challenges in this step such as the increased semantic ambiguities resulting from having multiple domains and the increased complexity of mapping different parts of the query to ontologies of different domains, if required. Tackling such challenges requires extra processing and more advanced query transformation techniques. Usually, this affects the system's performance (retrieval as well as runtime), and thus an inverse relationship between the system's performance and domain independence has been acknowledged [Kau07, ULL+07, DAC10]: the more specifically the system is oriented towards a domain application, the higher the performance it achieves.

In *closed-domain*, a single application domain is described and only queries requesting information in this domain can be answered. Semantic search systems in this category are either tailored to work with one *specific domain*, such as 'Libraries' [LM07], or are *domain-independent* and portable across different domains, though still operating in one domain at a time. Additionally, there are different levels of domain independence and portability: heavy customisation is sometimes required, as in the case of ORAKEL [CHHM07], while a balance between performance and easy customisation could be achieved, as in Querix [KBZ06], AquaLog [LPM05], and NLP-Reduce. The heavy customisation is usually due to a need for human intervention to manually adapt an automatically-generated lexicon to the new domain [CHHM07].

In *open-domain*, multiple application domains are covered and the posed queries may request information spanning these domains. While the source of information can be one or more predefined heterogeneous datasets, such as DBpedia, it is more often the open Web of Data, in order to take advantage of the huge potential offered for answering a wider range of users' queries. However, with this potential more challenges arise, especially in the open web scenario where the sources of information are usually undefined. Identifying the relevant data sources to answer queries, as well as disambiguating query terms and mapping them to the semantically equivalent terms from a large number of

[5] http://squin.sourceforge.net/

[6] The reader should not confuse these terms with their definition in other fields including databases and federated querying.

candidates in these data sources, are among the challenges facing systems operating in this category [DFJ+04, TOD07, LMU06, DAC10, HBF09].

The rest of this section discusses the necessary query transformations required for both closed-domain and open-domain systems from the perspective of the query approaches described in Section 3.3 (see Table 3.2 for a summary of systems and approaches). We structure the section in this way because the query format adopted influences, to a considerable degree, the kinds of query processing that are required, or indeed possible.

3.4.1 Formal Approach

Although formal queries are not suitable for non-expert users, as discussed in Section 3.3, the formal approach has the advantage of skipping the limitations and issues faced by other approaches in the process of query transformation. These include the complexity of parsing and understanding a NL query, and the lack of relations between entities usually found in the keywords-based approach [ZWX+07]. Another advantage is that the terms used in the given query are similar if not identical to those found in the underlying search space, and thus there is no need for mapping user terms to those terms. Finally, since the query is given in a format that can be directly used for execution, this approach obviates the need for generating several formal queries corresponding to the user's query.

3.4.2 Keywords-based Approach

Systems accepting keywords as input either use them directly to lookup their internal indexes or try to match them with terms identified in the underlying ontologies, which are then used in generating the corresponding formal query. As an example of the first approach, Sindice uses a literal inverted-index to lookup the keywords given in the search queries. The index contains an entry for each literal extracted from the documents, together with a list of the URLs for the corresponding documents. The keywords entered by the users are looked up in the inverted index, and the list of documents containing those keywords is returned to the user. Thus, the system applies no query transformation and the query input terms are used as they are. Similarly, Swoogle searches its internal indexes for the input keywords, and the SWDs[7] matching those keywords are returned in ranked order.

[LUM06] apply the second approach (match query terms with the underlying ontologies) in their system *SemSearch*. They identify three alternative semantic mappings to a keyword. Those are either concepts (e.g. the keyword 'publications' matches the concept 'publication'), relations between concepts (e.g. the keyword 'author' matches the relation 'has-author') or instances (e.g. the keyword 'Enrico' matches the instance 'Enrico-Motta'). An index containing all the semantic entities (classes, properties and instances) found in the underlying repositories is implemented and used in SemSearch. The labels of semantic entities are used as the index entries, since labels are usually more understandable to users and thus can be more relevant to match with their terms [LUM06].

[7] A document in a Semantic Web language such as OWL and RDF.

Using the keywords entered by the users, the system first searches the index to find all the semantic matches for each keyword. Those matches are then used to translate the user query into formal queries.

In SemSearch, the user can input the subject that is the expected type of the search results as in the example '`cities: universities England`' where the expected type is 'cities' (recall Section 3.3). For simple queries consisting of only two keywords, the authors defined nine query templates which cover all the possible match combinations for the two words. The templates are used to support the generation of the formal queries. An example is the search query '`news: Enrico Motta`'. The query will be matched with the combinations 'subject matches a concept' and 'keywords match an instance'. In this case, the expected search results are the instances of the matched concept that have relations with values matching the instance. For example, an instance that has a relation 'news_about' with value 'Enrico-Motta' would be selected among the results. To formulate the query, the system generates the query template corresponding to the chosen match combination. The number of generated queries depends on how many semantic matches each keyword has: if the keyword 'news' had two matches which are the concept 'news' and the relation 'news_about' and 'Enrico Motta' had only the instance match, the system would then generate two formal queries.

3.4.3 Natural Language Approach

Systems in this category employ different parsing techniques to understand the natural language query, and follow different strategies to generate the corresponding formal query. In principle, the process starts with parsing the query to identify different word forms, usually extracted from the query syntax tree generated by a parser. The second main step is to match those extracted terms with ones found in the underlying search space. The word forms can be treated in various ways in different approaches. For instance, one system would try to match nouns with concepts in an ontology, and verbs with relations between them. Another system would treat all the word forms equally and search for semantic matches regardless the type of the match. Finally, the identified matches are used to generate the corresponding formal query.

To illustrate, AquaLog, an example of *independent closed-domain* systems, uses a language processing tool – *GATE* [CMBT02] – to parse the natural language query and obtain a set of annotations for different word forms such as nouns and verbs. It also makes use of the Annotation Patterns Engine used within GATE to identify terms, relations and question indicators. AquaLog has a separate component called RSS that takes the output of the previous parsing step. Its goal is to output ontology-compliant query triples that represent the input query. Within RSS, proper names are mapped to instances in the search space using distance metrics [CRF03]. Classes and relations are identified and mapped to terms in the ontology, not based solely on string similarity techniques but also using synonyms obtained from WordNet[8] [Fel98] and a domain-

[8]WordNet is a large lexical database of English, developed at Princeton University. Nouns, verbs, adjectives and adverbs are grouped into sets of synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Each unique meaning of

specific lexicon. The RSS is also responsible for resolving ambiguities identified during the mapping process. It uses heuristics, as well as semantics inherent in the ontology, to deal with both structurally (related to the structure of the sentence) and semantically (related to the meaning of the words) ambiguous sentences. If RSS failed, AquaLog would seek help from the user in solving the ambiguity identified. Moreover, AquaLog provides a learning mechanism to improve the efficiency of RSS over use. Thus, when new items or relations are learnt they will be recorded for reuse in similar situations. This learning mechanism is also used to generate the domain-specific lexicon and thus there is no need for the application domain to be predefined, and no manual/human intervention is required. Therefore, the time taken to customise AquaLog to a new domain is negligible, making it highly portable [LPM05].

In Librarian [LM07], an example of *specific closed-domain* systems, a similar transformation process is performed starting with a linguistic preprocessing step to identify word categories (e.g. verb), lemmatise words and split the query into triples of the form 'subject, verb, object'. However, in the second step during ontology mapping, a domain-specific dictionary that was created previously is used as the lexicon instead of WordNet. This is worth noting since, as explained above, the degree of domain dependency affects the transformation process and its level of complexity, as well as the performance of the system. The advantages of this include the considerable reduction in dictionary size which in turn improves the system's performance (runtime). Additionally, it alleviates the problem of ambiguous interpretations that are usually returned for a given word by a generic lexicon such as WordNet [LM07], which would, in contrast, harm the performance. This is supported by the results of the evaluation carried out by [LM07] using 229 questions, among which 223 questions were solved correctly and for 86 of the questions, only one (the best match) answer was returned.

At the other end of the spectrum are the *open-domain* systems operating in an environment covering various domains. PowerAqua is an example that belongs to this category. It is the successor of AquaLog, and makes use of some of the components implemented in that system. Whereas the linguistic component of AquaLog, which carries out the parsing step, is still suitable for use in PowerAqua, the mainly syntax driven techniques used in the RSS in mapping user terms to ontology triples showed weakness and insufficiency. This is due to the fact that instead of applying this process with a few ontologies in a single domain, PowerAqua scales this up to the open web, handling multiple ontologies spanning various domains. The computational overhead of applying such techniques gets higher as the number of ontologies searched through increases, which together with the problem of the ambiguity that is more evident in multiple domains raised the need for PowerAqua to extend the steps and techniques done by the RSS component. PowerAqua tries to match the query with the ontology that covers most of the query terms. The query triple returned by the linguistic component may need to be restructured and transformed into sub-query triples to be matched with different ontologies. An example to show this situation is given by [LMU06] as 'Which researchers play football'. This query is translated by the linguistic component into

a word is presented by a synset: http://wordnet.princeton.edu

the triple '<researchers, play, football>', which needs to be restructured into the triples '<?, is-a, researcher>' and '<?, is-a, footballer>' and matched with the relevant ontologies that cover each triple separately. In a different situation, a query may generate matches with multiple candidate ontologies. Another example given by the authors is the query 'What is the capital of Spain'. This will identify matches with a geographic ontology containing 'capital-city' as a relation and 'Spain' as a country; a financial ontology with the terms 'capital' and 'Spain' as an instance of a country; and two other ontologies describing data about '*country statistics*' and '*flights information*'. This example makes evident the weakness and insufficiency of syntax-driven techniques in multiple-domain ontology mapping. Therefore, PowerAqua performs semantic analysis as an additional step to discard syntactically related terms that are not semantically equivalent to the query terms by applying sense-based similarity matching algorithms that make use of WordNet. WordNet is used to identify lexical relations such as meronymy and hypernymy between terms as well as using its indexes depth and common parent index to evaluate the distance between two concepts in a given input hierarchy, which helps in assessing their relatedness. Referring back to the query example, PowerAqua is able to exclude the three irrelevant ontologies and match the query terms with the geographical ontology to create the triple '<?, capital-city, Spain>'.

While AquaLog and PowerAqua accept free-form NL queries (represented as keywords, sentence fragments or full questions), Querix requires full English questions with restriction to the sentence beginnings (sentence must start with Which, What, How many, How much, Give me, or Does). Querix is another example of *independent closed-domain* systems and therefore operates in a similar manner to AquaLog: trying to map the user's query to a few ontologies or knowledge bases describing a single domain.

Unlike the above systems, QAKiS [CAC+12], a recently developed question answering system (currently limited to DBpedia) attempts to match phrases, rather than single terms, in an input query to ontology triples by relying on a repository of relational patterns. The intuition for this approach is to reduce the possibility of a wrong match as a result of a word-based match that is not guaranteed to capture the context around the word. The WikiFramework repository [MWB+11] used by QAKiS contains relational patterns extracted from Wikipedia. These patterns provide different lexicalisations for a specific relation. For instance, the relation `birthDate`(Person, Date) can be expressed by the pattern "`Person was born in Date`".

To match an input query to a relational pattern[9], the first step is to identify the *Expected Answer Type (EAT)* which is based on a set of predefined heuristics (e.g. "When" would be "Date" or "Time"). Then, named entities found in the query are identified using the Stanford named entity recogniser. If no entities were identified using this approach, then a search is performed on DBpedia to find matches for any proper nouns found in the query. If both approaches failed to identify any named entities, then the whole query is used to find the longest match with a DBpedia instance. The

[9]Currently, only questions containing a Named Entity (NE) that is related to the answer through one property of the ontology can be answered by QAKiS [CAC+12]

EAT and the named entity are then used to generate typed questions by replacing the query keyword by the supertypes of the EAT and the named entity by its type. To illustrate, the query "Who is the husband of Amanda Palmer?" would generate nine different typed questions from the EAT as `Person` and `Organisation` and the named entity *Amanda Palmer*, who has the concepts `MusicalArtist, Artist` and `owl:Thing` as its types. The most likely relation is identified based on string similarity comparison between the stems, lemmas, and tokens in the query and in the patterns. A maximum of five patterns are retrieved for each typed question which are then used to generate the final SPARQL queries. [LUCM13] notes that this approach of using a pattern repository can help in bridging the gap between user- and ontology- terms. However, limitations of this approach include scaling to other datasets as well as the need for at least one named entity to exist in the input query.

3.4.3.1 Polysemy and Synonymy

Polysemy and synonymy are two known problems in the wider language processing field. Polysemy – a single word form having more than one meaning – affects precision by causing false matches while synonymy – multiple words having the same meaning – affects recall by causing true matches to be missed [Voo93].

Polysemy: Depending on the query format used, a system might not need to address this problem. For instance, in systems using forms or graphs as well as those accepting formal queries, the user query contains the same terms found in the underlying knowledge base as was explained in Section 3.3. However, this is not the case for keywords-based and natural language approaches. One of the strategies to tackle this problem is to use the context of the query together with semantics of the ontology to understand and identify the correct sense of a word, as employed in FREyA [DAC10]. Another strategy used by Querix is to seek help from the user to clarify the ambiguity.

FREyA engages the user in solving the ambiguity, but only after trying to solve it using both query and ontology semantics as described earlier. For example, a query consisting of the word 'Mississippi' can be matched with either of the concepts 'State' or 'River' [DAC10]. The difficulty here is that even after using the semantics of the ontology, there is no way to understand the meaning intended by the user if the query lacks context. In such situations, FREyA will ask the user to clarify the ambiguity. Figure 3.10 shows the user engagement to clarify an ambiguity for a different query example. However, if the input query was 'Which rivers flow through Mississippi?' it will be able to solve this ambiguity without help from the user. The reason is that the relation 'flow through' is found to be between the concepts 'River' and 'State' and thus using the context of the query, 'Mississippi' would be correctly matched with the concept 'State'.

Unlike FREyA, Querix passes the responsibility of ambiguity clarification directly to the user and does not try to solve it first. The rationale behind this is to avoid implementing complex techniques to resolve ambiguities and to favour simplicity in the design and implementation of the system [KBZ06]. An example given by the authors is 'What

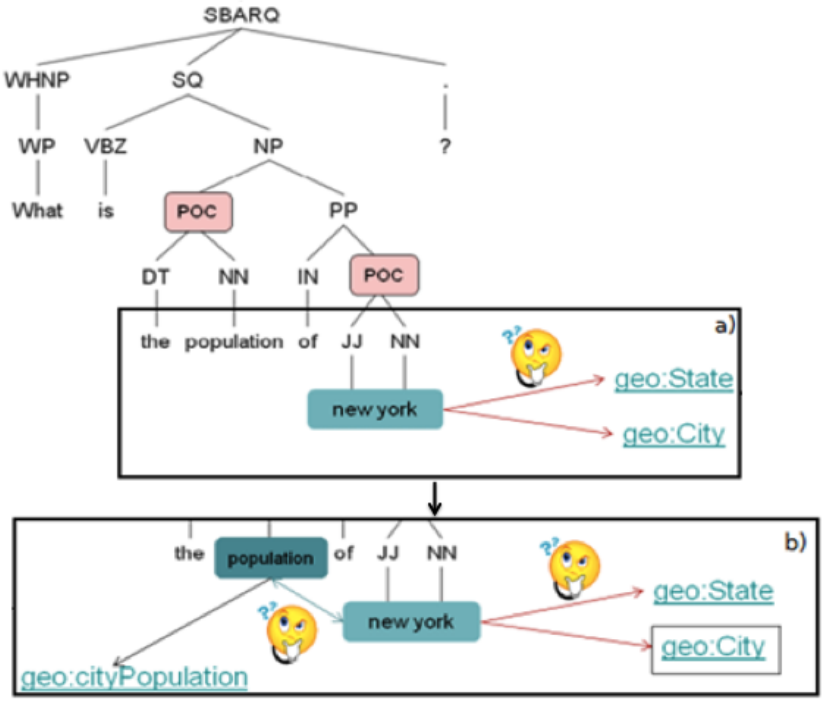


Figure 3.10: Validation of potential ontology concepts through the user interaction by FREyA [DAC10].

is the biggest state in the US'. The system retrieves synonyms for all the nouns, verbs and adjectives from WordNet, which would relate the term 'biggest' to 'size', 'number' and 'quantity'. Using those terms in addition to the main query term, the system tries to find matches in the ontology. It identifies the relations 'hasStateArea', 'hasStatePopulationDensity' and 'hasStatePopulation' as candidate mappings. The next step is to generate the corresponding formal queries for each of the mappings. Those queries are then shown to the user to choose the one that best describes their information need. For instance, if the user chooses 'hasStateArea', the system executes the corresponding formal query and returns the answer to the user, which in this case would be 'Alaska'. Figure 3.11 shows the formal queries generated for this example as presented to the user.

Synonymy: When different words describe the same meaning, the difficulty is to find results associated with all of the synonyms regardless of the one used in the query. *Query expansion*, which deals with this problem, enhances the recall by adding words to the query that are related in some sense to query terms. A number of query expansion techniques have been employed in the IR literature. One of those approaches is based on identifying useful or related terms to the query from the corpus/documents being searched and is usually referred to as *global analysis/technique* [Jon71, QF93, CR00]. In contrast, *local/retrieval feedback* [CRBG02] extracts such terms from top retrieved documents resulting for the original query [BS95, XC96, MSB98, CWNM02].

However, in semantic search, query expansion – if applied – is usually based on

extracting the related terms from domain-specific and/or generic ontologies [Kau07, LMU06, BMS07, MBH+09]. Some systems favour precision over recall and thus do not apply any query expansion techniques. This is because a term used for expansion can have different meanings in addition to the one it shares with the query term. This would add false matches to the query results, affecting precision.

To illustrate, Querix extracts synonyms for all the nouns, verbs and adjectives in the query from WordNet. The authors use the query example 'What are the population sizes of the cities that are located in California?' to show the synonyms returned by WordNet. The noun 'cities' will have the synonyms 'town', 'metropolis', 'urban center' and 'municipal', while those for 'California' are 'CA' and 'Golden State'. WordNet provides multiple synsets for the same word, each describing a different meaning. Having these different meanings – which can be from different domains – is one of the problems identified with using a generic lexicon for query expansion, since it can harm the retrieval precision by introducing false matches.

[KBZ06] do not explain how the cost function used to obtain the most appropriate synset for a given query term is implemented. In the same context, PowerAqua extends a query with synonyms, hypernyms and hyponyms obtained from WordNet for each query term. This decision was taken by the authors to maximise recall. Unlike Querix, which uses only the most appropriate synset for a given term, PowerAqua processes all of the synsets returned by WordNet which are retrieved during the query expansion phase. In the next phase following query expansion, the system tries to match a term or its lexical variations to ontology classes, relations or instances. The query example 'What is the capital of Spain?' given by the authors illustrates this as shown in Table 3.3.

Synonyms, hypernyms and hyponyms of all the synsets are used in the matching phase. For instance, 'Das-Kapital' was found as an instance of the concept 'book' in an ontology covering bibliographic information. The rationale is again for maximising recall, so as not to miss any candidate matchings that could be found when all the synsets are included.

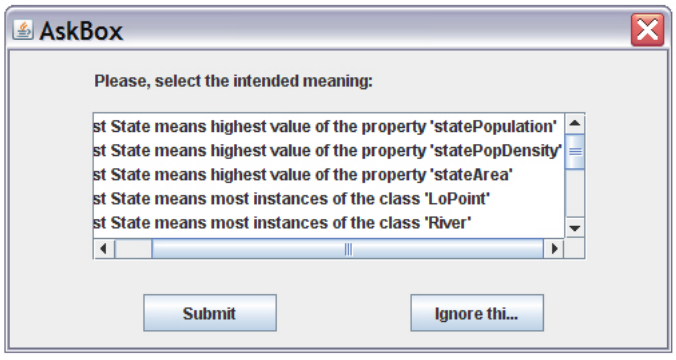


Figure 3.11: The Querix clarification dialog component [KBZ06]

Table 3.3: Lexical variations of the word "capital" as obtained from WordNet [LMU06].

Capital (glosses)	**Synonyms**	**Hypernyms**	**Hyponyms**
#1: assets available for use in the production of further assets	working capital	assets	Stock, venture capita, risk capita, operating capital
#2: wealth in the form of money or property	-	assets	endowment, endowment fund, means, substance, principal, corpus, sum
#3: a seat of government	-	seat	Camelot, national / provincial / state capital
#4: one of the large alphabetic characters used as the first letter	capital letter, uppercase, majuscule	character, grapheme, graphic symbol	small capital, small cap

3.4.4 Graph- and Form-based Approaches

Search systems employing either graph-based or form-based approaches for accepting a user query avoid the overhead of mapping user terms to the corresponding terms and/or relations used by the underlying ontology. They also side-step the syntactic ambiguities and the burden of applying complicated techniques to understand and transform a NL or keywords query. Moreover, they do not face the lack of relations or context that is usually encountered by systems employing the keywords-based approach.

In Section 3.3, we explained how a user can build their query using Semantic Crystal's graphical interface. When a user selects an element (concept or property), it gets presented on the SPARQL dashboard, found on the right side of the user's interface (see Figure 3.4). The system can execute a query when it is complete; it must consist of four elements known by the system as the *TORC approach*. The first element is the *Token(s)*, which represents the concepts in the ontology. The second is the *Output* which specifies the elements to be returned to the user in the results. This is used by the system to identify the classes and/or relations that need to be in the SPARQL SELECT statements. Next, the *Restriction* indicates the values given for any datatype properties that the system can use in the FILTER statements. Finally, the *Connection* is represented by the object properties, which are used to connect the query tokens. The corresponding SPARQL query gets iteratively constructed through the user's selections, and can be directly executed to return the required results.

3.4.5 Hybrid Approach

As seen in the previous sections, all the query input approaches discussed so far have been based on a single input approach (e.g. keywords). An alternative to this is the combination of two or more existing input styles. Each approach has specific advantages and capabilities in terms of its expressiveness (what queries are allowed), support during query formulation, as well as usability. They also have different limitations and face different challenges (e.g. the habitability problem faced by keyword- and NL-based approaches). The aim/objective of the hybrid approach is thus to try to produce a system in which any disadvantages/limitations of one approach are ameliorated by the

advantages of the other approach(es). For instance, using a NL approach could help in alleviating the tedium of a view-based approach by providing users with a faster and easier alternative for starting their search. Similarly, the view-based approach could support the NL one in overcoming the habitability problem by showing the users candidate matches for their input terms that they can then choose from. While [HV03] and [BCC+08] attempted to exploit the potential of this approach, work in this area is still limited and more studies are needed to understand the best ways to hybridise semantic search systems. However, it is important to investigate the difficulties arising from having two different query approaches, especially with respect to usability and learnability. Recent studies have shown that users can find the usability of the interface and support during query formulations (particularly for complex queries) challenging when using one query interface, let alone two [Kau07].

3.4.6 Summary

During query transformation, semantic search systems face various challenges such as the syntactic ambiguities or the lack of relations between given entities. Another major challenge is mapping the query terms to the equivalent ones in the available ontologies to generate the final/formal query. However, the need to address these challenges depends on the adopted format for the query input. The formal approach and the view-based approach do not face these challenges. In the former, the query can be directly executed against the underlying knowledge base, while in the latter the user selects from the concepts and relations found in the ontology and therefore no mapping is required. In closed-domain environments, the mapping is performed on a few ontologies describing a single domain, but this usually increases to a very large number of ontologies spanning multiple domains in open-domain environments such as the Web. The high number of available ontologies increases the complexity of the process and thus necessitates the use of advanced processing techniques including efficient blocking algorithms for filtering and reducing the number of candidate ontologies. Additionally, the heterogeneity of the data causes semantic ambiguities which in turn increase the difficulty of the mapping process. Among the common approaches to tackle this challenge are either to use the context of the query or to involve the users to help resolve any possible ambiguities.

3.5 Query Execution

In this step, semantic search systems generate the answers to the user's query and pass them to the next step, namely the *Results Presentation*. While some systems use the exact query terms to search their indexes and return matching documents (e.g. Swoogle and Sindice), others perform the *Query Transformation* step explained in the previous section to generate formal queries which are then executed against local or distributed triple stores to return answers.

Recall, semantic search systems operate in either a single closed-domain or an open-domain environment. Operating in the latter raises new challenges that need to be

addressed by these systems. The most important are scalability, performance (in terms of responsiveness) and the ability to provide up-to-date results. The two approaches for executing queries in an open domain – data warehousing and distributed query processing (DQP) – achieve different levels with respect to these aspects. For instance, while data warehousing is known to provide excellent response times, DQP has better chances in providing live up-to-date data. In addition to these common challenges, each approach faces additional ones resulting from its mode of operation. For example, there are many practical issues to be addressed in building and maintaining a data warehouse. For DQP, one challenge is in splitting and distributing queries to the right data sources and integrating their results.

The rest of this section reviews different semantic search approaches from the perspective of domain-dependence and how they tackle the above challenges.

3.5.1 Closed-domain Environment

As described in Section 3.4, systems in this category operate in a single application domain which could be either specific (e.g. Geography) or independent (e.g. Geography or Medicine). Hence, the main difference in this step is with respect to the underlying knowledge base(s) against which the user queries are executed. They are usually predetermined and integrated as part of the system in the first scenario (specific closed domain), while in the latter they are usually undefined and thus loaded each time the system is deployed in a new domain.

In principle, there are basic steps that are usually performed by any system in this category. Firstly, one or more knowledge bases describing the application domain are loaded into the system. These usually get enhanced and expanded with related terms (such as synonyms) from a domain-specific or generic lexicon. After that, a framework for ontology access and management (such as Jena[10]) is used to construct ontology models using the enhanced knowledge base. All the previous steps are often performed as a one-time task whenever the system is used within a new domain with a new knowledge base. Finally, a query execution engine (such as Jena ARQ) is used to execute the queries and the results are then passed to the next step (see Section 3.6).

For instance, Querix [KBZ06] and Panto [WXZY07] are domain-independent, and thus new knowledge bases are loaded when operating in a new application domain. As discussed in Section 3.4.3, for most of the NL-based approaches the user query is expanded using synonyms (and in some cases hypernyms and hyponyms) obtained from a lexicon for the query terms. In order to increase the matches between the query terms and the terms understood/used by the system, the same expansion process is used in the knowledge base enhancement step (mentioned above). WordNet is used by both Querix and Panto for this enhancement step. However, while Querix retrieves synonyms only for nouns and verbs, Panto retrieves them for all entities found in the knowledge bases. Ontology models are built in Querix (as in Semantic Crystal) using Jena which provides a programmatic environment for RDF, RDFS/OWL as well as SPARQL and includes

[10] Jena is a Java framework for building Semantic Web applications, see http://jena.sourceforge.net/index.html

a rule-based inference engine. In contrast, Panto uses Protégé[11], which is a free, open-source platform that provides users with a suite of systems to construct domain models and knowledge-based applications with ontologies. Another difference is the use of the Pellet reasoner[12] together with Jena to perform reasoning and infer additional triples from relationships such as the *Subclass*. While Jena contains its own query engine (ARQ) which is used by both Querix and Semantic Crystal to execute the generated SPARQL queries against the ontology models, the Protégé API lacks its own query engine and instead wraps a Jena model, meaning queries submitted to Panto are also executed by ARQ.

3.5.2 Open-domain Environment

In contrast, systems in this category operate in an open environment – ideally the whole Web of Data – spanning multiple domains. [HHK+09] described two different approaches to execute queries over the Web of Data. In the first approach, known as *data warehousing*, systems crawl the (semantic) Web to collect data, index it and store the results in some sort of a database, which is then used as the source for executing queries and retrieving results. In the other approach, which is based on *distributed query processing* (also known as *federated query processing*), systems parse and split the query into subqueries, determine the sources containing potential results for subqueries, and evaluate the subqueries against the sources directly [HHK+09].

3.5.2.1 Data Warehousing (DW)

The term *data warehouse* is traditionally defined as "a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process" [Inm05]. Another definition given by Kimball is "a copy of transaction data specifically structured for query and analysis" [KR02]. Data warehousing is a complex process covering all aspects of building, managing and querying a data warehouse. *Extraction, transformation* and *loading* are the main steps followed in building a data warehouse. Firstly, the warehouse designer selects the information sources containing the data of interest. Data from these information sources is extracted during the first step (Extraction). The next step (Transformation) involves transforming the data from the source format and language to the one used by the warehouse, as well as resolving any inconsistencies. Finally, the data is loaded into the warehouse (Loading).

Most of the semantic web search engines, such as Swoogle, SWSE, Watson and Sindice, apply data warehousing. Usually, multiple crawlers are used in this process for different tasks; for instance, to crawl documents within websites using some filtering constraints or to discover semantic links while parsing semantic web documents.

Although systems employing the data warehousing approach provide excellent query response times and address the problem of completeness as they index the Web, they

[11] http://protege.stanford.edu/
[12] http://clarkparsia.com/pellet

Table 3.4: Semantic search systems review for query execution

Systems	Closed-domain	Open-domain				
		Data Warehousing	Distributed Query Processing			
			DL	DS	SLI	ULO
AquaLog	√					
Corese	√					
DARQ					√	
FalconS		√				
FREyA		√				
Ginseng	√					
[HHK+09]				√		
K-Search	√					
Librarian	√					
LOQUS						√
Nlp-Reduce	√					
Ontogator	√					
Panto	√					
PowerAqua		√				
Pythia		√				
QAKiS		√				
Querix	√					
Semantic Crystal	√					
SemSearch	√					
Sig.ma		√				
Sindice		√				
SQUIN			√			
Swoogle		√				
SWSE		√				
TBSL		√				
Watson		√				

cannot provide up-to-date data. This is because the need to index this vast amount of information causes delays in getting the most recently updated data. There are also legal and technical difficulties in having copies of the data, such as restrictions by data providers or technical problems with huge datasets. Additionally, the resources overhead incurred by systems applying this approach (e.g. storage, computing power and technical personnel) is another limitation. Moreover, these systems need to handle issues, known in the IR community, with regards to managing a data warehouse. Some of these issues are: 1) change detection in the information sources and updating the warehouse to reflect the change; 2) addition or removal of information sources to and from the warehouse; 3) managing outdated data and 4) handling inconsistencies, duplicates and quality-related issues in the data.

The interested reader can refer to [DFJ+04, HHD+07, dBG+07] and [TOD07] for further information about data warehousing for the semantic web.

3.5.2.2 Distributed Query Processing (DQP)

One way to side-step most of the limitations of the data warehousing approach is to apply distributed query processing. The typical DQP involves query parsing and rewriting (splitting the query into parts that can be executed separately), source selection (identifying candidate data sources for answering the query parts) and finally query execution against those identified sources [HHK+09]. Recent research has focused on investigating the problem of source identification and selection within this approach. Four different approaches emerged to tackle this problem, namely: *direct lookup*, *data summaries*, *schema level indexes* and *upper level ontology*.

Direct Lookup (DL) is based on traversing RDF links and identifying data relevant to a query while executing it. Thus, no advanced knowledge is assumed about the data that might contain answers for a query. SQUIN [HBF09] adopts this approach and collects relevant data by looking up URIs given in a query. The retrieved data would contain more URIs that are dereferenced to find more relevant data. This would provide solutions for other parts of the query. This process is done continuously until all parts of the query are solved. The limitations of this approach include the need for initial URIs in the queries to start the link traversal, infinite link discovery, the retrieval of unforeseeably large RDF graphs, and URI dereferencing that takes unexpectedly long.

Data Summaries (DS) is regarded as a middle-ground between typical *data warehousing (DW)* and *distributed query processing (DQP)* approaches. Rather than indexing every item as in a typical DW approach, or working completely without an index as in a DQP one, it indexes a data summary that represents an approximate description of instance and schema level elements found in the data source. The interested reader can refer to [HHK+09] for information on building these data summaries.

Schema-Level Indexes (SLI) (adopted by [QL08]) is based on indexing classes and properties found in the data sources and using these indexes to identify relevant data sources for answering a query. It partially addresses the incompleteness of results faced by the direct lookup approach. However, it only addresses this problem partially since the index covers only schema-level elements found in the data sources and therefore, queries containing references to instances cannot be solved since instances are not indexed.

The previous approaches assume knowledge of the structure of the underlying ontologies or common LD vocabularies and datasets (e.g. foaf, skos[13]) by the users to formulate their queries. [JVY+10] argue that time and expertise are required for writing similar queries. Thus, they present a different method for querying LD, to overcome this limitation. Their method is based on having an *upper level ontology (ULO)* from which users can select concepts and relations to use in their queries. This ontology is mapped to datasets in the LOD cloud in which the mapping is used to translate and execute the user's query against the relevant datasets. Unfortunately, this approach has to deal with issues similar to the ones identified for data warehousing. These include the need for maintaining the upper ontology as opposed to a warehouse, the need for change

[13] http://www.w3.org/TR/skos-reference/

detection for the mappings between the ontology and the datasets as well as managing the ontology evolution to add or remove datasets as they change.

3.5.3 Summary

The Semantic Web in general and semantic search in particular have been gradually moving from focusing on closed-domains towards open-domains. The move was motivated by the huge potential offered by the emergence of Linked Data. The ability to reason over, collect and integrate information from numerous connected datasets offers new opportunities to answer millions of user queries. Both data warehousing and distributed query processing have been adopted in the literature to query the open Web of Data. Those in favour of the first approach (data warehousing) claim that it can achieve completeness and higher recall, since it crawls and indexes the Semantic Web rather than only specific data sources. Additionally, it can provide excellent query response times, since the data is stored in one place and can be queried with no need to wait to distribute the query or to search for relevant data sources. In contrast, those against data warehousing show the difficulties and challenges facing it [Wid95]. The ability to provide up-to-date data is usually a challenge, especially with the dynamic nature of Linked Data [HBF09, BHBL09, PHHD10, GS11]. Additionally, building and managing a data warehouse is resource intensive. These challenges and limitations have been driving more research in the area of distributed query processing. It attempts to find new data on-the-fly which overcomes the problem of the data becoming stale, and alleviates the need for the resources required to create and manage the warehouse. However, among the drawbacks of this approach is the increase in the response times, which can be caused by identifying relevant data sources [HHK+09, GS11], splitting and distributing the query, as well as URI dereferencing [QL08, MGSS10] which is applied in the *direct lookup* approach. Moreover, this approach faces a challenge with respect to the completeness of the results since only specific data sources are used to answer a query, and hence there is the possibility of losing relevant information [HBF09].

Both approaches mentioned above face other problems/challenges related to the openness and size of the Semantic Web. Among these are scaling up to the size of the Web whilst guaranteeing real-time response times, issues with data quality (noise, inconsistencies and varying-quality data sources) as well as tracking data-provenance. For instance, to be able scale up, common approaches include caching of data, or of queries and their results, in order to speed up query execution and responses. Additionally, [PHHD10] state that the variable quality of Linked Data is due to errors and inconsistencies which naturally arise in an open environment. Therefore, it is impractical to address this problem by correcting/validating every piece of information published on the Semantic Web (although there are efforts in this area by organisations like the Pedantic Web Group[14]) and thus more researchers have focused on investigating ways to track the provenance of data to be able to assess its quality. Most of the work in this area is concerned with adding meaningful metadata (data origin and size, date of publish,

[14] http://pedantic-web.org/

access methods, etc.) together with any published data [CSD+08, Har09, ACHZ09].

3.6 Results Presentation

In Section 3.3, we showed how different query approaches offered users different levels of support and usability. Similarly, results presentation – what to present and how to present it – can substantially affect how users perceive and evaluate a system's usability. The second part of this question (*how to present results*) usually depends on the task and use of the system. For instance, most of the Semantic Web search engines including Watson, Sindice and Swoogle have been mainly used by Semantic Web applications as entry points to locate documents that contain specific terms [BHBL09]. Therefore, they usually adopt the traditional approach of showing a ranked list of documents. However, unlike traditional IR systems, the graph structure of semantic web data means that a ranked list has limitations for displaying this kind of data in the most informative way, because it misses the – highly important – links.

The kinds of challenges facing presentation approaches vary according to the adopted presentation format, as do the challenges in answering the first part of the question (*what to present*). However, results ranking is a common, and critical, challenge for all formats. Results ranking highlights the most relevant results for a user's query. The scale of available data in the Semantic Web (which keeps increasing) necessitates search engines to apply ranking on the returned results (which can be in the range of thousands or millions of documents). Several studies have shown that users expect to find the best answers at the top of the results list; this directly influences their decision of clicking on a result [GJG04, JGP+05]. Additionally, approaches returning direct answers have to deal with challenges such as merging results gathered from different sources for the same (part of a) query, integrating results of different parts of a query (subqueries), as well as resolving similar instances and results cleaning for reducing redundancy in the answers.

The most common approach to presenting search results is a ranked list of Semantic Web documents. Users are familiar with this approach since it is adopted by most traditional search engines such as Google and Yahoo!. Each result item usually includes one or more of the following pieces of information representing the document: title, keywords, URL and a summary extracted from the document on the basis of relatedness/relevance to the query. Together, these are usually referred to as a *document surrogate* [Hea09, p.120]. The quality and characteristics of a surrogate affect users' experience of the search process in general, and their perception of the relevance of the associated document in particular. Examining these effects on IR has been the focus of several studies, e.g. [MW08, CADW07].

In the rest of this section, we will review different semantic search approaches with respect to the adopted presentation format and the different techniques followed to address the previously mentioned challenges.

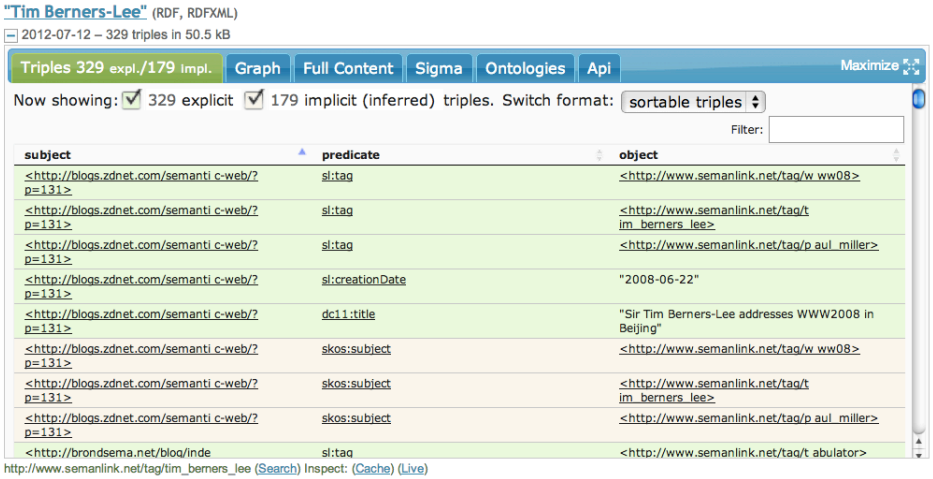


Figure 3.12: Part of the results returned by Sindice for the query 'Tim Berners Lee'.

3.6.1 Semantic Web Document List

Semantic Web search engines such as Watson, Sindice and Swoogle are regarded as gateways or entry points for the Semantic Web, which are used by Semantic Web applications and experts to find Semantic Web documents. Therefore, their results presentation format is targeted to the Semantic Web community and not optimised for the non-expert user. The notion of a *document surrogate* has thus been adopted in a different way from that in IR: semantic search engines discussed so far include different kinds of information in their surrogates. For instance, in addition to the normally-used title and URL of a resulting document, Sindice shows the number of triples found in the document and offers the ability to view these triples, an RDF graph of them or the ontologies used within the document. In contrast, FalconS opts to show the values for a set of predicates associated with the queried entity in the underlying data (e.g. type, label, etc.). To illustrate, Figures 3.12 and 3.13 show parts of the results returned by Sindice and FalconS for the query 'Tim Berners Lee', respectively .

As mentioned earlier, results ranking is crucial for (semantic) search. In this con-

Figure 3.13: Part of the results returned by FalconS for the query 'Tim Berners Lee'.

text, Sindice ranks sources[15] individually based on predefined metadata rather than global ranking. For each source, it computes the values of three pieces of metadata: 1) hostname, 2) external rank, and 3) relevant sources. It then calculates an unweighted average from these values. For the first one (hostname), a source gets a high value if it was the 'official' source of information about the resource (the source's hostname is the same as the resource's hostname). For the second one (external rank), a source gets a high value if its host was ranked high using traditional Web ranking algorithms. Finally, for the third one (relevant sources), a source gets a high value if it contains rare terms rather than common terms, a metric that [TOD07] relates to TF/IDF in IR [FBY92].

In Swoogle, [DFJ+04] introduced a ranking technique inspired by PageRank [PBMW99], which they call *OntoRank*. As opposed to PageRank, which is based on a random surfing model [PBMW99], OntoRank is based on a *rational* random surfing model [DFJ+04] which accounts for the type of links usually found between SWDs. [DFJ+04] claim that this is more appropriate for the Semantic Web, in which different weights can be assigned for following a link depending on its type. For instance, 'imports(A,B)' would give a high probability for following this link because B is a semantic part of A. However, this added step of differentiating links depending on their type is not sufficient for ranking SWDs. Ding et al. found that around half of all SWDs are not referred to by any other SWDs, and the majority of them are poorly connected, thus producing poor ranked results. Additionally, [AB05] note that a popular ontology does not necessarily indicate a good representation of all the concepts it covers.

3.6.2 Natural Language Answers

The process of finding information/answers in a list of documents can be laborious and time-consuming. It usually requires the user to examine some or all of these documents to identify the ones of interest, then they have to locate, organise and integrate the required information to generate the final answers for their queries. This process can take many iterations, especially if large result sets are returned. Search engines try to assist the process by improving ranking algorithms to return the best results first. State of the art systems are attempting to move beyond simple lists of documents by providing the information sought by users (i.e., including an additional interpretive step). To do this, they provide users with the natural language answers to their queries rather than a list of documents. However, returning direct answers raises new issues and challenges in the results generation phase. These include merging query results coming from different sources, integrating results of different parts of a query (subqueries) in order to form the final answers, and results cleaning (e.g. removal of duplicates). Indeed, results ranking is still required in this approach. However, rather than ranking all documents containing matches for a user's query terms, this is usually performed on a sub-query basis, as will be explained next.

Usually, the process of generating the final answers is performed in two steps. The

[15] [TOD07] differentiates between a resource such as 'http://eyaloren.org/foaf.rdf#me' and a source (e.g. an RDF document or a SPARQL endpoint), that provides information about that resource, such as 'http://eyaloren.org/foaf.rdf'.

first step is responsible for merging and integrating the results of all parts of a query answered by one or more data sources. For example, in PowerAqua, for queries including *and/or* such as 'who are the professors affiliated with the University of Sheffield and who went to ISWC2011?', the instances resulting from answering the first part: 'professors affiliated with the University of Sheffield' are integrated with the instances resulting from answering the second part: 'professors who went to ISWC2011' to form the final answers for the query. A more complex scenario is when there is a need to resolve a specific part of a query to use the answer in another part. Consider the query given by [LMU06]: 'What are the homepages of the researchers working on the Semantic Web?'. This requires identifying the list of researchers working on the Semantic Web and then using this list to answer the first part of the query: to find the researchers' homepages. One of the major challenges in this step is *schema matching* which "aims at identifying semantic correspondences between two schemas, such as database schemas, XML message formats, and ontologies" [DR07, p.857]. In order to integrate results from different data sources, systems need to be able to identify and match equivalent concepts and properties. There is a large body of related work on ontology matching/mapping [CSH06, ES07] supported by the *Ontology Alignment Evaluation Initiative (OAEI)*[16]. One approach that is being adopted within semantic search is based on using generic lexicons or upper ontologies to identify similarity [GY04, RB01, KS99]. For instance, [LMU06] uses WordNet to identify such similarities between different ontologies. In this approach, two concepts/properties are considered similar if they are found in the same synset (e.g. 'human' and 'person') or one of them is a hypernym/hyponym of the other.

This step also involves ranking of the different ontology matches (Onto-Triples) generated for the same sub-query triple. As discussed in Section 3.4, PowerAqua generates sub-query triples for each query. Recall the query 'Which researchers play football' generates the sub-query triples '<`?, is-a, researcher`>' and '<`?, is-a, footballer`>'. If a sub-query triple produces multiple matches with the underlying ontologies (Onto-Triples), these are ranked using three different algorithms. The first algorithm uses WordNet to compute semantic similarity distances between the candidate Onto-Triples. Onto-Triples are then ranked according to their popularity: how semantically-similar they are to other Onto-Triples. The second algorithm performs ranking according to the quality of the Onto-Triple, which depends on the type of the mapping linking it to the sub-query triple. For instance, exact mappings are ranked the highest. Finally, the third algorithm ranks answers according to their popularity in terms of the number of ontologies from which they were extracted.

After integrating the information to form complete answers for a query, the second step is to resolve similar instances to guarantee non-redundant answers. This process is usually known as *Instance Matching* [CFMV11] – also referred to as *record linkage* [FS69b, Win99] or *entity resolution* [KR10] – and identifies different instances representing the same real world object. Ideally (according to the Semantic Web standards), these instances would be linked using the 'owl:sameAs' property. When this link is not

[16] `http://oaei.ontologymatching.org/`

provided, it is usually difficult to establish this similarity, especially when these instances have different labels, different information or even different URIs when gathered from different data sources. In principle, instance matching is based on computing a similarity degree between the instances. Two instances are then considered similar if this degree reaches a predefined threshold [BN09]. Some of the techniques adopted to address this problem are based on applying string similarity algorithms (e.g. string edit distance and cosine similarity) [EIV07, BM03]. Others use the properties associated with the instances to compute the similarity degree [FS69a, HPUZ10]. In this case, instances are considered similar if the similarity between their properties reaches the specified threshold.

The open nature of the Semantic Web naturally leads to data characterised by variable levels of quality (which thus will always have errors and noise) and heterogeneity (published by different sources and spanning different domains). Providing information about the provenance of different pieces of data returned as results to a user query is thus hugely important to assess data quality and reliability [PHHD10, HZ10, OZG+11]. It is worth noting that this problem is not limited to a specific results presentation approach (e.g. list of documents or NL answers). However, it is discussed in this section since it is more essential/critical to provide such provenance information when presenting 'answers' that are a result of extraction, reasoning, filtering and integration of data. In contrast, when returning URLs of documents to users, they can identify the document's source of information from the hostname (e.g. BBC) or navigate to the given URL to get more information to help them assess the data quality. For instance, PowerAqua shows the ontologies that were used to generate the answers (origin/source) and the mappings that were found between the ontology and the query terms (akin to reasoning)[17]. The usefulness of this information, however, can depend on the 'type' of the user. For instance, knowing that a specific answer for a Geography query was extracted from 'GeoNames' can provide confidence for a Semantic Web expert but not necessarily to a non-expert user. Furthermore, this information can cause confusion for the latter. Therefore, a separation between the results shown to an expert and those shown to a non-expert user would be useful, as they have different requirements and knowledge and thus require different views. FREyA [DAC10] follows this approach and presents the natural language answer separately from the ontology concepts and relations which are given at the bottom of the page for the expert user.

3.6.3 Entity Description

Unlike the previously discussed systems which return individual results in isolation, an alternative is to integrate those results with the information contained in them to create a rich and comprehensive view of the returned entities. This is more akin to exploring the Web of Data than to searching and finding answers to specific queries. Systems following this approach (also called mashups) usually apply it to an entity within the

[17] It should be emphasised that PowerAqua's interface was only intended to be used by the developer rather than by end users

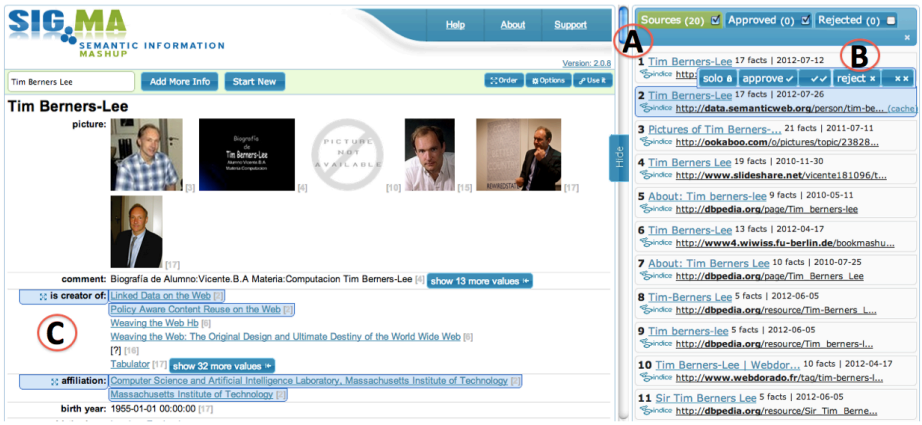


Figure 3.14: Example of a Sig.ma profile. (A): sources contributing to a profile; (B): approving or rejecting sources; (C): values highlighted when hovering over the source from which they were extracted.

Web of Data (e.g. 'Person', 'Location', 'Event', etc.), rather than to queries in which users seek specific answers to their information needs.

Sig.ma [TCC+10], a mashup built on top of Sindice[18], creates information aggregates called Entity Profiles, or the 'sig.ma' of an entity and provides users with various capabilities to organise, use and establish the provenance of the results. Figure 3.14 shows part of the sig.ma for 'Tim Berners Lee'. It contains his pictures, his birth year, some of his publications as well as his affiliations. The rest of the sig.ma contains more information such as his homepage, location and links to some of his colleagues. In addition to this, Sig.ma allows users to have more control over the results. For instance, users can see all the sources contributing to a specific profile (Figure 3.14, highlight A) and approve or reject certain ones (Figure 3.14, highlight B), thus filtering the results. They can also check which values in the profile are given by a specific source: they are highlighted when the user hovers over the source (Figure 3.14, highlight C), thus checking provenance of the results. While Sig.ma supports merging separate results, it also allows users to view ones extracted from specific sources.

However, Sig.ma does not try to do any automatic disambiguation for the query terms: it gives the user an interactive way to remove false or unwanted results. Putting the responsibility on the user for disambiguating each and every value returned can be impractical, especially for a large number of false or unwanted results. Additionally, although giving the users the ability to filter the results according to specific data sources could be seen as providing them with more control, it requires knowledge about such data sources and ontologies used within the Semantic Web (such as 'swdf', 'opencyc' or 'foaf'). This is therefore suitable only for Semantic Web experts who have this knowledge.

Referring back to deciding *what to present* as results for users' queries, Sig.ma performs two major steps in answering this question while creating these entity profiles

[18] http://sindice.com/

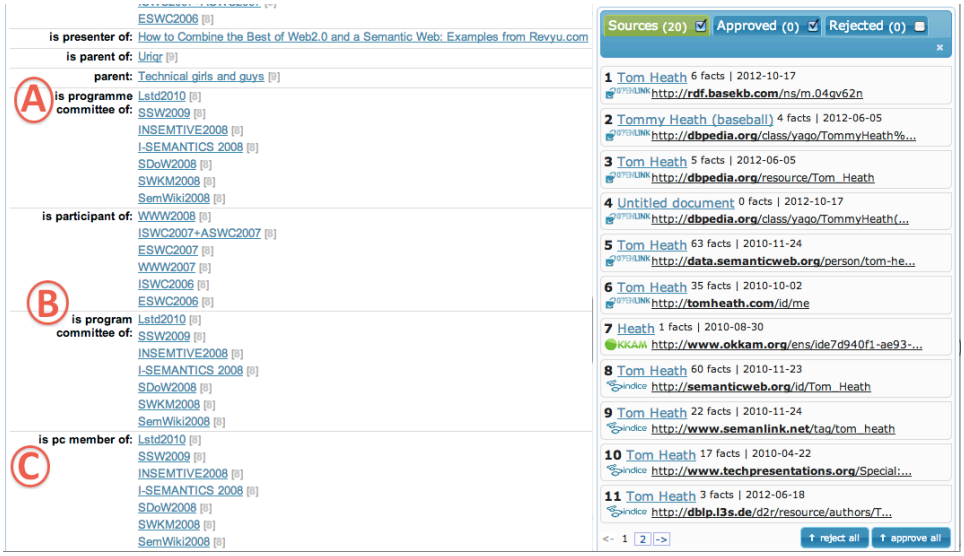


Figure 3.15: Different properties with equal values found in a sig.ma. (A): is programme committee of; (B): is program committee of; (C): is pc member of.

described above, namely: data gathering and data consolidation. In the first step, information about entities found in the query is retrieved from different data sources (currently set to 25 sources) using Yahoo Boss and Sindice. This data is then clustered into *resource descriptions* about distinct entities. For instance, a resource description about 'Tim Berners Lee' would have his URI as the subject or object of each triple included in the description. All descriptions (coming from different data sources) about the same entity are then scored and ranked according to their similarity with the query keywords (considering both RDF literals and words in URIs) [TCC+10]. Only descriptions with similarity scores above certain threshold are passed to the next step.

In the second step (data consolidation), descriptions included from the previous step are merged into a single entity profile. The challenge here is to consolidate a large chaotic list of properties gathered from various data sources into a simpler list that is meaningful to the user. An important task to reduce redundancy is to identify similar properties. This is done by first identifying names of the properties found in the profile. Thus, the *local part* of the URI of each property is converted into a readable name consisting of space-separated words. For improved readability, these names are further processed based on predefined heuristics; for instance, by removing "has" in "has title". A final step of data transformation and consolidation replaces each property name with a suitable match from a list of 50 manually-compiled *preferred terms*. For example, "page", "homepage", "url", and "website" are all replaced by the preferred term "web page".

Despite this attempt to provide meaningful and non-redundant results for users, descriptions returned by Sig.ma usually contain different properties that have equal values. Figure 3.15 shows the sig.ma of 'Tom Heath' in which the three properties 'is programme committee of' (Figure 3.15, highlight A), 'is program committee of' (Figure 3.15, high-

light B) and 'is pc member of' (Figure 3.15, highlight C) show the conferences in which Tom Heath was a member of the programme committee. These properties could be from the same data source or from different ones; in both cases, equivalent properties should be identified as such and linked through equivalence relationships. However, this is only in theory. In practice, datasets in the linked open data cloud are loosely coupled, lacking the required links [JHS+10, PKA10].

Although there is still much work to be done in publishing linked data to reduce these problems, it is inevitable that these issues will remain to some degree when working in a large and open environment. Therefore, it is desirable that applications attempt to help overcome the impact of these problems since they can affect user satisfaction and systems' usability.

In this context, it is worth mentioning Google's *Knowledge Graph*[19] which is built by collecting information about objects in the real world and connecting these objects in an attempt to introduce improvements to the search performance and experience. One of these improvements is adding context to the search results by augmenting them with 'related' information. Google combines what other people have found useful for a specific search query – referred to by Google as *collective human wisdom* – with the information found in its knowledge graph to bring more meaningful results for its users. For example, a search for "Leonardo da Vinci" would return, in addition to the traditional ranked list of documents, information about him (such as his birth and death date) as well as about related topics. The latter would include information about some of his paintings such as "the Mona Lisa", or about other *related* painters (related could be, for instance, in profession or era, among other criteria defined by Google) such as "Michelangelo".

3.6.4 Graphical Visualisation

Presenting search results as a list of documents or natural language answers is limited in most cases, as it does not put the results within context. Every document or answer is considered on its own, which might not be enough in some situations. For instance, clustering results into meaningful categories can support the user in understanding the organisation of the results, how they are related and how they belong to different contexts. Also, plotting search results on charts or maps can provide effective overviews of the results, which in turn helps in understanding their structure or easily comparing them according to suitable criteria. In general, graphical visualisation of the results can assist the users in discovering more information, and making useful findings that otherwise would be missed. Semantic data visualisation is an important emerging research area addressing novel means of exploring and browsing data [DR11].

K-Search is an example of systems adopting this approach for results presentation. As discussed earlier in Section 3.3, K-Search is an implementation of a hybrid search strategy in which either or both of semantic search and keyword-based search are applied, depending on the availability of metadata in the queried parts. K-Search provides different presentations of the search results; the traditional ranked list of documents as

[19] http://www.google.com/insidesearch/features/search/knowledge.html

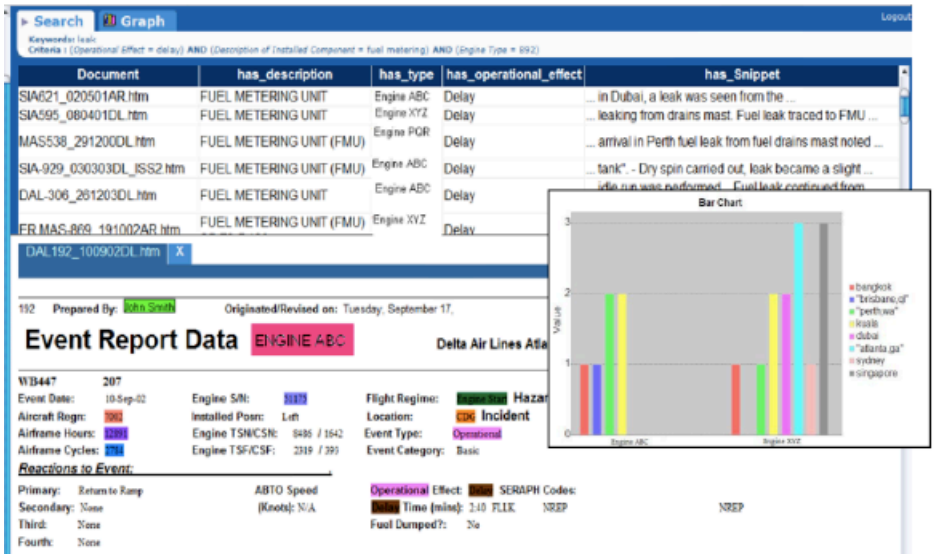


Figure 3.16: "K-Search interface showing the list of documents returned (centre top), an annotated document and a graph produced from the results (image modified to protect confidential data)" [BCC+08].

well as a graphical presentation of the results. As shown in Figure 3.16, the resulting documents are listed in order in the middle of the interface. If metadata-search was used in the query, the matching values in each document are also given in the list. K-Search uses the structured data to provide a graphical view of the results. As shown in Figure 3.16, the plotted graph groups the results with respect to a specified concept chosen by the user. Also, every bar in the chart can be clicked to identify the documents that belong to this specific cluster. Finally, users of K-Search can select either a pie or a bar chart to view the results depending on their preferences.

Table 3.5 shows the different results presentation approaches adopted by the semantic search systems reviewed above.

3.6.5 Summary

Providing direct natural language answers as opposed to a list of documents is an attempt to reduce the amount of time and effort required by users in the information seeking process. However, the former is limited in putting results in context since it is usually restricted to the NL representation of each result item which is also shown in isolation. One approach to address this limitation is based on graphical visualisations (e.g. clustering results or plotting them on charts or maps) which provide users a wider understanding of the results. An alternative is integrating information about entities given in a query, which provides a comprehensive view of the results. A challenge facing all the above approaches concerns results refinement and cleaning. Due to noise and the variable quality of data found on the Semantic Web, it is common to observe multiple properties in the same data source that refer to the same real world property.

For instance, the three properties 'prop:birthDate', 'onto:birthDate', 'prop:dateOfBirth' found in DBpedia are all used interchangeably to refer to the date of birth of a person. Identifying these equivalent properties and merging them is important in order to return non-redundant and hence more meaningful results to the user. Usually, resolving conflicts while merging the values of these properties is even more difficult than identifying them. This situation occurs when these properties, although referring to the same real world object, have different values. The difficulty of resolving these inconsistencies drove the adoption of solutions based on engaging the user in selecting the appropriate value. A related challenge – concerning entities as opposed to properties – is to identify similar instances (instance-matching), which also reduces redundancy in the results. Furthermore, approaches performing integration of results from different data sources need to address schema matching, which is a major research problem. Additionally, efficient ranking techniques are required to distinguish the most relevant results for a user's query from the overwhelming amounts of available data.

3.7 Summary

This chapter has described the different definitions found in literature for the term *semantic search* and the ones covered within this thesis. It has also presented a review of the literature in semantic search which was focused on the main aspects by which these approaches differ; namely, the format for the input query, the underlying search mechanisms as well as the results presentation format. Additionally, it has discussed the most relevant challenges facing each approach and different solutions adopted to tackle them. For instance, the variance in the levels of support offered by different input query formats (such as keywords - or view - based) for users during query formulation, and also in their expressiveness and flexibility raises the question of understanding which approach is suitable and provides the most satisfaction for end users and why. In an attempt to answer this question, Chapter 7 presents a usability study investigating how end users perceive the usability of these different approaches.

Table 3.5: Semantic search systems review for results presentation

Systems	**Results Presentation**			
	Ranked list of documents	Natural Language answers	Entity description	Graphical visualisation (e.g. graphs, charts, maps)
Affective Graphs		√		
AquaLog		√		
Corese		√		
DARQ		√		
FalconS		√		
FREyA		√		
Ginseng		√		
[HHK+09]		√		
K-Search				√
Librarian	√			
LOQUS		√		
Nlp-Reduce		√		
Ontogator				√
Panto		√		
PowerAqua		√		
Pythia		√		
QAKiS		√		
Querix		√		
Semantic Crystal		√		
SemSearch	√			
Sig.ma			√	
Sindice	√			
Smeagol		√		
SQUIN		√		
Swoogle	√			
SWSE		√		
TBSL		√		
Watson	√			

Chapter 4

Evaluation of Information Retrieval and Semantic Search Systems

4.1 Introduction

Evaluation is highly important for designing, developing and maintaining effective Information Retrieval (or search) systems as it enables the success of an IR system to be quantified and measured [JÏ1]. This can involve evaluating characteristics of the IR system itself, such as its retrieval effectiveness, or assessing consumers' acceptance or satisfaction with the system [Tau55]. For decades, the primary approach to IR evaluation has been system-oriented (or batch-mode), focusing on assessing how well a system can find documents of interest given a specification of the user's information need. One of the most-used methodologies for conducting IR experimentation that can be repeated and conducted in a controlled lab-based setting is test collection-based evaluation [Rob08, San10, JÏ1, Har11]. This approach to evaluation has its origins in experiments conducted at Cranfield library in the UK, which ran between 1958 and 1966, and is often referred to as the 'Cranfield approach' or methodology [Cle91]. Although proposed in the 1960s, this approach was popularised through the NIST-funded Text REtrieval Conference (TREC) series of large-scale evaluation campaigns that began in 1992 and stimulated significant developments in IR over the past 20 years [VH05].

However, despite the many benefits that come from the organisation of evaluation activities like TREC, the semantic search community still lacks a similar initiative on this scale of activity. Indeed, [HHM+10] note that "the lack of standardised evaluation has become a serious bottleneck to further progress in this field". In recent years evaluation activities have been organised to address this issue, including the SemSearch Challenge [HHM+10], the SEALS semantic search evaluations [WRE+10, WGCT11], the QALD open challenge [UCLM11] and the TREC Entity List Completion task [BSdV10]. However, these initiatives have yet to experience the level of participation shown by eval-

uation exercises in other fields.

Although the focus of this chapter is to provide a background on evaluations and review related work on evaluating semantic search systems, I believe that there is much to learn from the IR community about evaluation as both share the goal of helping users locate relevant information. Additionally, how to conduct IR system evaluation has been an active area of research for the past 50 years and the subject of much discussion and debate [Sar95, Rob08, Har11]. This is due, in part, to the need of incorporating users and user interaction into evaluation studies, and the relationship between results of laboratory-based vs. operational tests (Robertson1992). Therefore, most of the background in this chapter is from IR literature.

The rest of the chapter is organised as follows. In Sections 4.2 – 4.4, literature regarding IR evaluation is discussed, including information covering important aspects such as the different evaluation paradigms, selection of the document collections and queries, the judgment process and evaluation measures. In Section 4.5, existing semantic search evaluation initiatives are reviewed with respect to important aspects such as the datasets used or the evaluation measures adopted.

4.2 Approaches to IR Evaluation

Evaluation is the process of assessing the 'value' of something, and evaluating the performance of an IR system is an important part of the development process [Sar95, Rob08]. For example, it is necessary to establish to what extent the system being developed meets the needs of the end user, to show the effects of changing the underlying system or its functionality on system performance, and to enable quantitative comparison between different systems and approaches. Success might refer to whether an IR system retrieves relevant (compared with non-relevant) documents; how quickly results are returned; how well the system supports users' interactions; whether users are satisfied with the results; how easily users can use the system and the effort demanded from the user (intellectual or physical) [CMK66, p. 4].

How to conduct IR system evaluation has been an active area of research for the past 50 years and the subject of much discussion and debate [Sar95, Rob08, Har11]. This is due, in part, to the need of incorporating users and user interactions into evaluation studies, and the relationship between results of laboratory-based vs. operational tests [RHB92]. Evaluation of retrieval systems tends to focus on either the system or the user. [Sar95] distinguishes six levels of evaluation objectives, not mutually exclusive, for information systems, including IR systems:

1. The *engineering level* deals with aspects of technology, such as computer hardware and networks to assess issues such as reliability, errors, failures and faults.

2. The *input level* deals with assessing the inputs and contents of the system to evaluate aspects such as coverage of the document collection.

3. The *processing level* deals with how the inputs are processed to assess aspects such as the performance of algorithms for indexing and retrieval.

4. The *output level* deals with interactions with the system and output(s) obtained to assess aspects such as search interactions, feedback and outputs. This could include assessing usability, for example.

5. The *use and user level* assesses how well the IR system supports people with their searching tasks in the wider context of information-seeking behaviour (e.g. the user's specific seeking and work tasks). This could include, assessing the quality of the information returned from the IR system for work tasks.

6. The *social level* deals with issues of impact on the environment (e.g. within an organisation), and could include assessing aspects such as productivity, effects on decision–making and socio–cognitive relevance.

Traditionally, in IR evaluation there has been a strong emphasis on measuring system performance (levels 1-3), especially retrieval effectiveness [Rob08, Har11]. The creation of standardised benchmarks for quantifying retrieval effectiveness (commonly known as *test collections*) is highly beneficial when assessing system performance. This is because the test collection enables the absolute assessment of individual systems, as well as the relative assessment and comparison amongst a group of systems [Rob08, San10, CS13]. However, evaluation from a user-oriented perspective (levels 4-6) is also important in assessing whether a system meets the information needs of its users by taking into account characteristics of the user, their context and situation, and their interactions with an IR system, perhaps in a real-life operational setting. This includes, for example, assessing the usability of the search interface or measuring aspects of the user's information-searching behaviour (e.g. a user's satisfaction with the search results or the number of items viewed/saved) [Bor09, Kel09].

4.3 System-oriented Evaluation

One of the first and most influential proposals for system-oriented evaluation was based upon the Cranfield methodology [Cle60]. The Cranfield approach to IR evaluation uses test collections: re-useable and standardised resources that can be used to evaluate IR systems with respect to the system. Over the years, the creation of a standard test environment has proven invaluable for the design and evaluation of practical retrieval systems by enabling researchers to assess in an objective and systematic way the ability of retrieval systems to locate documents relevant to a specific user need. Alternative approaches to conducting system-oriented evaluation include comparing results from multiple systems in a side-by-side manner [TH06] and A/B testing, where a small proportion of traffic from an operational system is directed to an alternative version of the system and the resulting user interaction behaviour compared [MRS08].

4.3.1 Evaluation using Test Collections

The main components of a standard IR test collection are the document collection (Section 4.3.2), statements of users' information needs, called topics (Section 4.3.3), and an

assessment for each topic about which documents retrieved are relevant, called relevance assessments (Section 4.3.4). These, together with evaluation measures (Section 4.3.5), simulate the users of a search system in an operational setting and enable the effectiveness of an IR system to be quantified. Evaluating IR systems in this manner enables the comparison of different search algorithms and the effects of altering algorithm parameters to be systematically observed and quantified. The most common way of using the Cranfield approach is to compare various retrieval strategies or systems, which is referred to as *comparative evaluation*. In this case the focus is on the relative performance between systems, rather than absolute scores of system effectiveness.

Evaluation using the Cranfield approach is typically performed as follows: (1) select different retrieval strategies or systems to compare; (2) use these to produce ranked lists of documents (often called *runs*) for each query; (3) compute the effectiveness of each strategy for every query in the test collection as a function of relevant documents retrieved; (4) average the scores over all queries to compute overall effectiveness of the strategy or system; and (5) use the scores to rank the strategies/systems relative to each other. In addition, statistical tests may be used to determine whether the differences between effectiveness scores for strategies/systems and their rankings are significant. This is necessary if one wants to determine the 'best' approach. In the TREC-style version of the Cranfield approach, there is a further stage required prior to (2) above, whereby the runs for each query are used to create a pool of documents (known as *pooling*) that are judged for relevance, often by domain experts [JB77]. This produces a list of relevant documents (often called *qrels*) for each query that is required in computing system effectiveness with relevance-based measures (e.g. precision and recall).

Test collection-based evaluation is highly popular as a method for developing retrieval strategies. Benchmarks can be used by multiple researchers to evaluate in a standardised manner and with the same experimental set up, thereby enabling the comparison of results. In addition, user-oriented evaluation, although highly beneficial, is costly and complex and often difficult to replicate. It is this stability and standardisation that makes the test collection so attractive. However, there are a number of limitations to test collection-based evaluation due to its abstraction from reality [IJ05, pp. 6-9]. Test collections experiments make a number of assumptions [Voo02]: that the relevance of documents is independent of each other; that all documents are equally important; that the user's information need remains static; that a single set of judgments for a query is representative of the user population; and that the lists of relevant documents for each query are exhaustive.

By modifying the components of a test collection and evaluation measures used, different retrieval problems and domains can be simulated. The original and most common problem modelled is ad hoc retrieval (the situation in which an IR system is presented with a previously unseen query). However, test collection-based evaluations have also been carried out on tasks including question answering, information filtering, text summarisation, topic detection and tracking, and image and video retrieval. Further information about the practical construction of test collections can be found in [San10, CS13].

4.3.2 Document Collections

IR systems index documents that are retrieved in response to users' queries. A test collection must contain a static set of documents that should reflect the kinds of documents likely to be found in the operational setting or domain. Although similar in principle to traditional IR, in the case of semantic search a knowledge base (e.g. RDF data) is typically the document collection (the term *dataset* will be used throughout this section to refer to both document and data collections). Datasets can be constructed in different ways. For example, they can be *operationally derived* or *specially created* [SJVR76]. Additionally, a dataset can be *closed/domain-specific* (e.g. those used in biomedicine) or *open/heterogeneous* and spanning multiple domains (e.g. the web). In addition, datasets can differ in *size* and *type of data*. [GNC10] discuss various issues around collecting datasets to form TREC-style test collections for evaluating visual information retrieval systems.

In IR, examples of document collections include the *Cranfield 2* collection [CMK66] that covered a single domain - aeronautics - and consisted of 1400 documents which were all research papers written in English. An example of a more recent collection is the *ClueWeb09*[1] collection of web pages used in the TREC Web track. It was operationally derived (crawled from the Web) in 2009, spans various domains, and consists of more than 1 billion web pages in 10 languages. Other examples of collections in IR include ISILT [KD72], UKCIS [BWVS74], MEDLARS [BBG72], and the different datasets used in TREC tracks[2].

In semantic search the geography dataset that forms part of the Mooney NL Learning Data [TM01] has been used in several studies [TM01, Kau07, DAC10]. It was specially created in 2001 and covers a single domain - geography. It consists of around 5700 pieces of information (RDF triples) published in English. An example of a larger dataset used in semantic search evaluations is DBpedia [BLK+09]. It is an extract of the structured information found in Wikipedia that was operationally derived and created in 2009. It covers various domains, such as geography, people and music and consists of around 1.8 billion RDF triples in multiple languages, such as English, German and French. Other examples of datasets used in semantic search studies include SWDF [3], BTC-2009[4] and Sindice-2011 [CCP+11].

4.3.3 Topics

In system-oriented evaluation, IR systems are evaluated for how well they answer users' search requests or queries. In the case of ad hoc retrieval, the test collection must contain a set of statements that describe typical users' information needs. These might be expressed as queries that are submitted to an IR system, questions, visual exemplars or longer written descriptions. TREC uses the notion of a 'topic', which typically consists of three fields: query (typically a set of keywords), title (a short sentence or

[1] http://lemurproject.org/clueweb09/
[2] http://trec.nist.gov/data.html
[3] http://data.semanticweb.org/
[4] http://km.aifb.kit.edu/projects/btc-2009/

phrase describing the search request) and description (a description of what constitutes a relevant or non-relevant item for each request). Topics will vary depending on the search context being modelled. For example, topics for an image retrieval system may consist of visual exemplars in addition to a textual description [GC07, MÏ0]. An example of a topic from the TREC-9 Web Track [VH00] is the following:

```
Number: 451
What is a Bengals cat?
Description: Provide information on the Bengal cat breed.
Narrative:
 Item should include any information on the Bengal cat
 breed, including description, origin, characteristics, breeding
 program, names of breeders and catteries carrying bengals.
 References which discuss bengal clubs only are not relevant.
 Discussions of bengal tigers are not relevant.
```

The selection of realistic and representative topics is an important aspect of creating the test collection. The effectiveness of an IR system is measured on the basis of how well the system retrieves relevant items in response to given search requests. Typically, a range of topics will be chosen that test various aspects of an IR system. Criteria that may be used to select queries could include the following: the type of query (e.g. informational, navigational or transactional in the case of web search [Bro02]), the length of the query, the language of the query, and whether the query contains spelling mistakes, named entities or other features (e.g. temporal constraints). Additional factors that may be considered are the number of items retrieved for a given query, the number of relevant items, and diversity of topics or subjects covered by all queries. Ultimately, the goal for topic creation is to achieve a natural, balanced topic set accurately reflecting real world user statements of information needs [PB01].

There are various ways of obtaining typical search requests that may form the basis of topics. For example, analysing query and clickstream logs from operational search systems, utilising the findings of existing user studies, involving domain experts in the topic creation process, and conducting surveys and interviews amongst target user groups. Practically there will be a trade-off between the realism of queries and control over the testing of features for the search system being evaluated [Rob81]. With respect to the number of queries required to effectively test an IR system, research has suggested that 50 is the minimum for TREC-style evaluations [Voo09]. However, results have also shown that making fewer relevance judgments over greater numbers of queries leads to more reliable evaluation [CPK+08].

By way of example, the *Cranfield 2* test collection made use of 221 topics created by the authors of a number of papers selected from the document collection representing the papers' research questions. On the other hand, the *MEDLARS* [Lan68] test collection included 300 actual requests submitted to the 'National Library of Medicine' through its Medical Literature Analysis and Retrieval System. Similarly, examples of datasets used in semantic search studies include the Mooney geography dataset containing 1000 sentences collected from students as well as from real users through a Web interface. In comparison, real queries obtained from logs of two search engines (Yahoo! Search and Microsoft Live Search) were used in the SemSearch evaluation campaigns

(see Section 4.5.2).

4.3.4 Relevance Assessments

For each topic in the test collection, a set of relevance judgments must be created indicating which documents in the collection are relevant to each topic. The notion of relevance used in the Cranfield approach is commonly interpreted as *topical relevance*: whether a document contains information on the same topic as the query. In addition, relevance is assumed to be consistent across assessors and static across judgments. However, this is a narrow view of relevance which has been shown to be subjective, situational and multi-dimensional [Sch94]. Some have speculated that the variability with which people judge relevance would affect the accuracy with which retrieval effectiveness is measured. However, a series of experiments were conducted to test this hypothesis [Cle70, Voo98] with results showing that, despite there being marked differences in the documents that different assessors judged as relevant or non-relevant, the differences did not substantially affect the relative ordering of IR systems being measured using the different assessments.

Various studies and authors in IR literature have used different terms interchangeably in relation to the notion of relevance, including *relevance, pertinence, situational relevance, logical relevance, system relevance, user relevance, satisfaction, usefulness, topicality, aboutness, subject-relevance* and *utility* [KBLP55, CK67, Coo71, Kem74, Wil78, SEN90, Miz98, Sar07]. Broadly speaking, discussions of relevance are centred around the notions of *system relevance* and *user relevance* [Vic59b, Vic59a]. Here, we use the term *system-relevance*, which is usually related to *topicality, logical-relevance, subject-relevance* and *aboutness*, to refer to whether a document or a piece of information is related to/about a given query or topics. In contrast, the term *user-relevance* is used in relation to *pertinence, situational relevance, satisfaction, utility* and *usefulness* to refer to the subjective usefulness/appropriateness of a document or a piece of information to an information need as perceived by the end user of an IR system.

To assess relevance, different scales have been used. The most popular of these is binary and graded relevance scales. When using a *binary relevance* scale, a document is judged as either relevant or non-relevant; in the case of *graded relevance* a document is judged for relevance on a scale with multiple categories, e.g. *highly relevant*, *partially relevant* or *non-relevant*. [Rob81] argues that relevance should be treated as a continuous variable and hence, different levels of relevance should be incorporated in an evaluation model. Therefore, researchers have attempted to experiment with non-dichotomous relevance scales [Cua67, Eis88, Jan93, SGB98, TSV99]. The study by [TSV99] showed that a graded-relevance scale with seven points led to the highest levels of confidence by the judges during their assessments. The additional benefit of using graded relevance scales is that a wider range of system effectiveness measures, such as *discounted cumulated gain (DCG)* [JK02], can be used (see also Section 4.3.5). In recent years, this use of ordinal relevance scales together with the appropriate measures have become more common. For instance, a three-point relevance scale together with DCG as a measure were used

in the TREC Entity Track 2009. Similarly, the SemSearch evaluation (see Section 4.5.2) used a three-point relevance scale and a normalised version of DCG as the evaluation measure.

There are various ways of gathering the relevance assessments. For example, in TREC, the common approach used is the pooling technique, in which the top n results from the different IR systems under test are gathered for each topic and aggregated to form the required pool of results for judging.

This assumes that the result lists of different IR systems are diverse and therefore will bring relevant documents into the pool. The relevance assessors then go through the pool (or a sample of the pool) and make relevance judgments on each document which can then be used to compute system effectiveness. Documents which are not judged are often categorised as not relevant. This technique has been used in different tracks at TREC [VH99, VH05]. Alternative approaches to gathering relevance assessments include simulating queries and relevance assessments based on user's queries and clicks in search logs [ZK10].

An issue with pooling is the completeness of relevance assessments. Ideally, for each topic all relevant documents in the document collection should be found; however, pooling may only find a subset. Approaches to help overcome this include using results lists from searches conducted manually in the pool of documents for assessment, or supplementing the sets of relevance judgments with additional relevant documents discovered during further manual inspection. Generating complete sets of relevance judgments helps to ensure that when evaluating future systems, improvements in results can be detected. The effects of incomplete relevance assessments, imperfect judgments, potential biases in the relevance pool and the effects of assessor domain expertise in relation to the topic have been investigated in various studies [Zob98, BV04, YA06, BCYS07, BTC+08, KHZ08].

Generating relevance assessment is often highly time-consuming and labor intensive. This often leads to a bottleneck in the creation of test collections. Various 'low-cost evaluation' techniques have been proposed to make the process of relevance assessment more efficient. These include approaches based on focusing assessor effort on runs from particular systems or topics that are likely to contain more relevant documents [Zob98], sampling documents from the pool [APY06], supplementing pools with relevant documents found by manually searching the document collection with an IR system, known as Interactive Search and Judge or ISJ [CPC98], simulating queries and relevance assessments based on user's queries and clicks in search logs [ZK10] and using crowdsourcing [AM09, Kaz11, CLY11].

4.3.5 Evaluation Measures

Evaluation measures provide a way of quantifying retrieval effectiveness [MRS08, CMS09]. Together, the test collection and evaluation measures provide a simulation of the user of an IR system. For example, in the case of ad hoc retrieval, the user is modelled as submitting a single query and being presented with a ranked list of results. One as-

sumes that the user then starts at the top of the ranked list and works their way down, examining each document in turn for relevance. This, of course, is an estimation of how users behave; in practice they are often far less predictable. There are also further complications that must be considered. For example, research has shown that users are more likely to select documents higher up in the ranking (*rank bias*); measures typically assume that no connection exists between retrieved documents (*independence assumption*); and, particularly in the case of Web search, a decision must be made regarding whether to count duplicate documents as relevant or previously seen.

Although many properties could be assessed when evaluating a search system [CMK66], the most common approach has been to measure retrieval effectiveness. The most well known measures are *precision* and *recall* [KBLP55]. Precision measures the proportion of retrieved documents that are relevant; recall measures the proportion of relevant documents that are retrieved. Precision and recall have been widely used in IR evaluations as the main evaluation measures. However, as *set-based* measures, – which treat results as an unordered set or list – their use has been long criticised for not taking into consideration the ranking of results. Therefore, *ranked-based* evaluation measures have been introduced in the literature to overcome this problem. Some of the commonly used ones are described next.

4.3.5.1 Binary-Relevance Measures

When used for assessing relevance of the results, the following measures consider a result item to be either relevant or non-relevant, with no levels in-between.

4.3.5.1.1 Precision@k

Measures the number of relevant results found in the top k results (also known as the cutoff value) returned by an IR system for a specific query. It credits an IR system for ranking more relevant results in higher positions and has therefore been commonly used in assessing the performance of search engines while estimating a cutoff value as the number of results (documents or answers) users examine (often 10 or 20 are used). However, one major limitation for this measure is that the choice of this cutoff value influences the results. For instance, if the chosen value (e.g. 20) is higher than the number of relevant documents (e.g. 10) for a specific query, then this measure – precision@20 – would never reach 1 even for a system that retrieved each and every relevant document. This could be misleading and affecting the reliability of an evaluation.

4.3.5.1.2 R-Precision

In order to overcome the limitation discussed above with respect to the cutoff value, R-precision was created, in which R refers to the number of relevant documents for a specific query. The use of an unfixed/changeable cutoff value guarantees that a precision of 1 can be achieved. When this measure is used, precision and recall are of equal values since both of them are calculated as 'number of relevant documents retrieved / overall number of relevant documents (R)'.

4.3.5.1.3 Mean Average Precision (MAP)

This is a very frequently used measure in IR and semantic search evaluations which gives an overall figure of the systems' performance (in terms of precision). MAP is explained/calculated as the arithmetic mean of average precision values for a set of queries. These average precision (AP) values are equal to the areas under the precision-recall curves for the queries, and, therefore, the MAP value takes ranking into consideration.

4.3.5.1.4 Mean Reciprocal Rank

[KV00] defined this measure in order to evaluate the performance of IR systems in retrieving a specific relevant document, also known as a *known-item search*. It is calculated as the mean of the reciprocal ranks (RR) for a set of queries. For a specific query, RR is the reciprocal of the rank where the first correct/relevant result is given. For instance, the RR for a query where the relevant required result is given at rank 3 is 1/3. Although this measure is mostly used in search tasks when there is only one correct answer [KV00], others used it for assessing the performance of query suggestions [MBH+09, AKN+11], as well as ranking algorithms in particular [DAC10] and IR systems [Voo99, Voo03, MRV+03] in general.

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{4.1}$$

4.3.5.2 Graded-Relevance Measures

Although the above measures are very commonly used both within IR and semantic search evaluation, their main limitation is that they must be used with binary-relevance scale. As discussed in Section 4.3.4, this scale is insufficient when comparing IR systems with respect to their performance in retrieving documents with different levels of relevance. Therefore, other measures that can be used with graded-relevance judgments were introduced in literature to overcome this limitation. The rest of the section describes some of these measures that are widely adopted in both communities.

4.3.5.2.1 Direct Cumulated Gain (CG)

[JK00] based this measure on the observation that "highly relevant documents are more valuable than marginally relevant documents". Therefore, the more relevant a retrieved document is (with higher relevance grade), the more gain the evaluated IR system achieves. This gain is accumulated for the documents and thus the CG is calculated according to Equation 4.2, in which $G[i]$ is the relevance value of the document at position i.

$$CG[i] = \begin{cases} G[1] & \text{if } i = 1 \\ CG[i-1] + G[i] & \text{otherwise} \end{cases} \tag{4.2}$$

4.3.5.2.2 Discounted Cumulated Gain (DCG)

The direct cumulated gain (CG) does not account for ranking: differences in the ranking of documents do not change its value. To account for ranking and based on their second observation: "the greater the ranked position of a relevant document (of any relevance level) the less valuable it is for the user, because the less likely it is that the user will examine the document", [JK00] defined DCG with a discounting function to reduce the credit given for lower-ranked results. This function is chosen as the log of a document's rank in the list of results. DCG is calculated according to Equation 4.3. Again, $G[i]$ is the relevance value of the document at position i. Finally, the log base b can be adjusted as required; for instance, to have high discounts for Web search users who are interested in getting the most relevant results as highly ranked as possible.

$$DCG[i] = \begin{cases} G[1] & \text{if } i = 1 \\ DCG[i-1] + \frac{G[i]}{\log_b(i)} & \text{otherwise} \end{cases} \tag{4.3}$$

4.3.5.2.3 Normalised Discounted Cumulated Gain (NDCG)

Similar to how precision@k is influenced by the chosen cutoff value (k), the CG and DCG measures are influenced by the number of relevant documents for a query. This limitation prevents the comparison of different (D)CG values for different queries. This is tackled in the NDCG in which the DCG values are normalised with respect to an *ideal result list*. To calculate the NDCG, the DCG values are divided by the equivalent ideal values (those in the same position). As illustrated by [JK02], if the (D)CG vector V of an IR system is $< v_1, v_2, ..., v_k >$, and the ideal (D)CG vector I is $< i_1, i_2, ..., i_k >$, then the n(D)CG vector is given by

$$normVect(V, I) =< v_1/i_1, v_2/i_2, ..., v_k/i_k > . \tag{4.4}$$

4.3.5.2.4 Expected Reciprocal Rank

The main advantage known for the cumulated gain- measures described above is that they account for both the rank and the relevance level of a retrieved document. However, they do not account for previous documents found in the list of results and how they affect the relevance/usefulness of the current document. In contrast to this *simple position* model which assumes independence between relevance of documents found in a ranked result list, a *cascade* model is one in which the relevance of one document is influenced by the relevance of documents ahead in the result list. [CZTR08] showed that the latter better explains web search users' behaviour: "users view results from top to bottom and leave as soon as they see a worthwhile document". Therefore, [CMZG09] proposed the *expected reciprocal rank (ERR)* which is based on the cascade model in an attempt to have a more accurate measure for users' satisfaction. The ERR, at which a user is satisfied and examines no more documents, is calculated according to Equation 4.5[5] in which n is the number of documents in the ranking.

[5]The reader can refer to [CMZG09] for a detailed explanation of this equation.

$$ERR = \sum_{r=1}^{n} \frac{1}{r} P(user\_stops\_at\_position\_r) \tag{4.5}$$

All the above measures have been used in different studies and for different purposes with many contradicting claims for which measure is the best to assess retrieval performance. The choice of the evaluation measure depends on several aspects such as the scale of relevance adopted, the number of queries available as well as the number of results considered/assessed for each query, and also the purpose/goal of the search task [BV00]. For instance, precision@k would be an appropriate choice to assess IR systems helping users who are more concerned with and will only assess the top k results [Hul93, TS07]. Numerous research tried to assess and compare these measures against each other, especially with respect to their stability and discrimination power. [TsB94, BV00, VB02] and [Sak06] showed that precision at a fixed level of rank (P@k) was usually found to be the least discriminating with the highest error rates among other measures such as R-Precision and mean average precision (MAP). This is mainly due to the influence of the choice of the cutoff value on the results, as was explained above. Although [SZ05] confirmed this finding, they showed that when taking the assessor effort into consideration, P@10 is much more stable than MAP since it required only around 14% of the assessor effort required to calculate MAP. When comparing R-Precision and MAP, [TsB94] concluded that MAP had a higher discriminating power, a similar finding by [BV00]. However, the latter showed that the two measures had almost equal error rates. Fewer studies have investigated the stability of graded relevance-based measures (such as nDCG and ERR) or compared them with the binary-relevance- based ones. [Sak07] found that nDCG was as stable and sensitive as MAP while [RC10] found that the first is more stable when a small number of queries is used. When tested with query set sizes from 5 to 30, the authors showed that MAP results could be misleading since the worse ranking was sometimes statistically significantly better.

4.4 Interactive/User-oriented Evaluation Approaches

Evaluation from a user-oriented perspective is also important in assessing whether a system meets the information needs of its users and to obtain a more holistic view that incorporates users directly in the retrieval process [TS89, Sar95, HH97, Su92, Sar95, Voo02, IJ05]. To complement batch-mode system-oriented evaluations, various studies have been carried out from an Interactive Information Retrieval (IIR) perspective, such as those in TREC in the 1990s [Ove01, KL07] along with many others [Su92, Dun96, KB96, Xie03, Pet08, Hea09]. In this approach to evaluation, real users are required to use an IR system, perhaps in a controlled lab-based environment, and their interactions with the system are recorded, along with their feedback on the system and information about their individual characteristics (e.g. age, cognitive abilities, etc.). The remainder of this section discusses important aspects that should be addressed in conducting a user-oriented evaluation, such as the criteria to be assessed

(e.g. usability, utility, relevance, efficiency and user satisfaction), the choice of data collection methods, evaluation measures and issues related to the experimental setup.

4.4.1 Criteria and Measures

The goal of an IR system is to assist a specific user in fulfilling their information needs. Therefore, simply evaluating in terms of retrieval effectiveness (e.g. using precision and recall) is insufficient. Indeed, various studies have showed that user's satisfaction and success at performing a search task does not always correlate with high retrieval effectiveness [Hit79, Su92, Tag97, HTP+00, TH01, Her02, HV08]. In part, this is because users are able to adapt to poorly performing systems. Additionally, other factors influence the user's satisfaction with the search results, e.g. their domain knowledge and expertise; aspects of retrieved results such as its quality, language or authoritativeness; the presentation of search results; as well as the usability of a search system user interface. Criteria and measures concerned with how well users achieve their goals, their success and their satisfaction with the results have been used to evaluate IR systems. Measured aspects include efficiency, utility, informativeness, usefulness, usability, satisfaction and success, as well as quantifying search time, number of queries, success and error rate, response time, learning curve and user knowledge pre- and post-search as evaluation measures.

Many forms of criteria and associated evaluation measures have emerged in the literature of user-oriented IR. [Kel09] identifies four basic measures: (1) *contextual* that capture characteristics of the subject and tasks undertaken (e.g. age, gender, familiarity with search topics); (2) *interaction* that capture aspects of the user-system interaction (e.g. number of queries issued, number of documents viewed, etc.); (3) *performance* that relate to the outcome of users' interactions (e.g. number of relevant documents saved, precision, nDCG, etc.); and (4) *usability* that capture evaluative feedback from subjects (e.g. satisfaction, suggestions, attitudes, etc.).

The term *relevance* has been vaguely and inconsistently used in IIR literature, similar to its use in IR literature more generally. [Vic59b, Vic59a, Tau65] and [Soe94] used it to refer to the degree of match between a document and a question (how much the document can help the user with information about the question) as judged by the user. In contrast, [Fos72, Kem74, GN66] and [GN67] distinguished between *relevance* as a notion similar to system-relevance where this degree of match or relation between a document and a question is assessed by an external judge/expert and *pertinence* to refer to the user-relevance in which the assessment can only be performed by the real user with the information need represented in the question. Often, measures adopted in user-oriented studies are those which account for non-binary, subjective relevance assessments usually given by real users. These include the cumulated gain measures (CG, DCG, NDCG) presented earlier; the *relative relevance* [BI98] and *ranked half-life* [BI98] which are proposed specifically for IIR; as well as the binary-based measures: *expected search length* [Coo68] and *average search length* [Los98] proposed earlier in literature.

4.4.1.1 Relative Relevance (RR)

As mentioned in Section 4.3.4, relevance terminologies and definitions have been long debated in IR. These debates have been centred around two notions: system-relevance (also referred to as objective relevance) and user-relevance (referred to as subjective relevance). In an attempt to bridge the gap between both notions and evaluate the retrieval performance of an IR system with respect to both of them, [BI98] proposed the relative relevance measure. It is based on calculating an association between the two kinds of relevance based on Jaccard measure. It also helps understanding, for instance, if one IR system outperforms another only when evaluated objectively but is not as good when evaluated subjectively from a user's perspective. However, it does not take into account the rank of the retrieved document.

4.4.1.2 Ranked Half-Life (RHL)

In contrast to RR, this measure is more related to the ESL and ASL (described below) since it evaluates an IR system with respect to its ability to position relevant documents high in a ranked result list. It is defined as the position at which half of the relevant documents are retrieved. [Los98] explains that the advantage of having this median ranking is that its increase indicates that more highly-relevant documents were ranked at the top of the result list while its decrease indicates that these relevant documents were ranked in low or scattered positions in the result list. However, [JK02] argues that this is a downside of the measure, similar to ASL, since it is affected by outliers: relevant documents ranked at low positions.

4.4.1.3 Expected Search Length (ESL)

[Coo68] criticised most of the IR traditional measures based on precision and recall, especially for not being able to report a single measure of a system's performance and for not accounting for the user's need while evaluating the system's performance: "most measures do not take into account a crucial variable: the amount of material relevant to his query which the user actually needs". Therefore, he proposed the ESL to provide a single measure for assessing an IR system's performance in helping the user finding the required amount of relevant documents while reducing the effort wasted in examining irrelevant documents. It is calculated by finding this number of irrelevant documents that appear before the user retrieves his required 'K' relevant documents. Specifying the value of 'K' was often mentioned as a downside for this measure, especially since it differs according to the user and the query.

4.4.1.4 Average Search Length (ASL)

The ASL is the expected position of a relevant document in a ranked result list. It is a related measure to the ESL since it similarly takes into account the user's effort wasted in examining irrelevant documents and credits an IR system for positioning relevant documents high in the list to reduce this effort. However, both ESL and ASL are

criticised for allowing only binary-relevance assessments and thus they do not account for the degree of relevance of a document, unlike the cumulated gain measures.

4.4.1.5 Efficiency

The ISO standard defines efficiency as the "resources expended in relation to the accuracy and completeness with which users achieve goals" [ISO98]. IIR focuses on the interaction between users and IR systems, and their evaluations focus on assessing the success of users in achieving their goals and answering their information needs through this interaction. Therefore, efficiency of the users in this process has been one of the main evaluation criteria studied in IIR. While some work investigated the degree of correlation between efficiency and the user's overall success or satisfaction, others proposed and examined different measures that can be used to assess efficiency. The most common measures of efficiency are time and effort-based measures since both can indicate the user's efficiency in achieving a specific goal with a specific IR system. [SKCT87, Su92, Su98, TKP04] and [JGH03] used search time (also referred to as completion time) to measure efficiency. This is usually the time from when a user starts a specific search task till its end. In contrast to measuring user-efficiency, response time has been used to measure system-efficiency [DS97, JGH03]. Additionally, user effort has been measured in different ways: in the number of search terms used in a specific task [SKCT87]; in the number of commands used [SKCT87]; and in the number of queries issued to complete a specific task [Su98]. [TKP04] has also used the number of completed tasks in a time period or a session – which can be seen as the inverse of the search time – to measure efficiency. Interestingly, [Su92, DS97, Su98] and [JGH03] found that efficiency of an IR system was an important factor that influenced users' overall success and satisfaction. While Su and colleagues found that time was more influencing, Johnson reported that users in his study related efficiency to the required effort rather than to the task time.

4.4.1.6 Learnability

Learnability, used interchangeably with the term *ease of learning*, is an important criterion of usability that focuses on the ease of learning how to use a system or an interface. [Sha86] describes learnability as the relation of performance and efficiency to training and frequency of use. [Nie93] discusses how learnability can be measured in terms of the time required for a user to be able to perform certain tasks successfully or reach a specified level of proficiency. A similar definition is given by [Shn86] as "the time it takes members of the user community to learn how to use the commands relevant to a set of tasks". Nielsen argues that learnability could be seen as the most fundamental usability attribute since, in most cases, systems need to be easy to learn. [TA10] agree to this and argue that measuring usability in a one-time evaluation might be misleading since the use of some applications/systems requires frequency, and therefore assessing learnability would be essential.

Learnability has received a fair amount of research in literature, some of which fo-

cused on assessing learnability as a usability criterion while others investigated how it is affected by different factors (such as interface design). While some of this work focused on *initial learnability* (referring to the initial performance with the system), others looked at *extended learnability* (referring to the change in performance over time) [GFA09]. For example, [HEE+02] studied the learnability of two hypermedia authoring tools (HATs) as perceived by academics. Subjects' answers to a set of Likert scale-based questions and their feedback (recorded during the sessions) were used to investigate learnability issues. In [Par00], learnability of two different methods of interaction with databases was compared using similar measures which are based on subjective data (such as questionnaires and users' feedback). [Jen05] assessed the learnability of searching two university Web sites by asking students of the first university to search the other site and vice versa. In contrast to the previous studies, efficiency-based measures, including success rate (number of tasks performed correctly) and the time required to perform the tasks, were used to assess learnability. Additionally, [RM83, WJLW85, DBW89, HEE+02] showed that learnability and usability are congruent.

4.4.1.7 Utility

While [Fos72, Kem74, GN66] and [GN67] tried to differentiate between *relevance* and *pertinence* in an attempt to account for the subjectivity of relevance assessments, [SKCT88, Coo73a, Coo73b] and [Soe94] argued that utility was a more appropriate measure for evaluating IR systems and their ability to support users in their search tasks. [Soe94] explained that a document has utility if it is pertinent (relevant as perceived by the user) and also contributes to the user knowledge in the context of the query (by providing information that was previously unknown). Utility is usually adopted as a measure of usefulness and worth of the answers provided by the IR system to its users. The argument for using utility, as opposed to relevance or pertinence, is that a document that has information about the user query does not have to be 'useful' for the user since this depends on other aspects. For instance, the user might already know this information or it can be in a language or format that is not understood by the user. Additionally, the clarity, reliability and credibility of a document also affect its usefulness [Soe94, Coo73a, Coo73b]. There has long been debate over the best ways to assess this criterion, since it can be difficult to be quantitatively measured. Therefore, the most common ways have been through the use of questionnaires gathering users' answers on different questions chosen by researchers. For instance, [SKCT88] included questions such as "On a scale of 1 to 5, what contribution has this information made to the resolution of the problem which motivated your question?" in order to understand the degree of informativeness and usefulness of a document.

4.4.1.8 User Satisfaction

Both [Coo73a] and [SKCT88] used the terms *utility* and *satisfaction* equally to refer to the overall value of a search result or a document to the user. Similarly, [TCA77, Su92, CLCS92, Dra96, Su98] and [DLB00] used satisfaction as a multidimensional measure

to evaluate IR systems from users' perspectives. Satisfaction by a single or a group of search results or documents is very subjective and depends on various factors related to the user (e.g. knowledge or personal preferences), the IR system (e.g. responsiveness, interface or aesthetics) and indeed the results (e.g. completeness, accuracy or format). Similar to utility, satisfaction is often measured using questionnaires which address the factors mentioned here. In assessing users' satisfaction with libraries, [TCA77] included factors such as the output, the library as a whole, its policies and the interaction with the library staff. Other factors commonly used in studies include completeness of search results [Su92, Su98]; interface style [CLCS92, Hil01, MRS08]; interaction with the system [CLCS92, JRGH07, MRS08]; response time [Kel09]; features and functionality of the system [JRGH07]; ease of query formulation [DLB00] and ease of use [Hil01]. [Coo73a, SKCT88] and [Dra96] argued that satisfaction is the ultimate measure to evaluate IR systems. A counter argument by [Soe94] and [BV85] explained that utility – in contrast to satisfaction – evaluates IR systems with respect to their main functionality which is to help users find 'useful' information that can be used to answer their needs.

4.4.2 Experimental Setup

"An experiment is an examination of the relationship between two or more systems or interfaces (independent variable) and a set of outcome measures (dependent variables)" [HY07]. A user-oriented evaluation approach tries to involve real users in the assessment process of the IR system, take into account real information needs and adopt appropriate user-oriented criteria and measures. In a very broad sense, this is usually an experiment in which real users are recruited to assess a number of IR systems performing specific search tasks with respect to predefined criteria and in which different forms of data are usually collected (e.g. interaction logs, post-task questionnaires, etc.).

A general framework to guide user-oriented evaluation for all kinds of scenarios (not just for IIR) is known as DECIDE [PRS02]. This represents the following stages: (1) *D*etermine overall goals that the evaluation addresses; (2) *E*xplore specific questions to be answered; (3) *C*hoose the evaluation paradigm and techniques; (4) *I*dentify practical issues (e.g. the selection of participants and tasks for IIR); (5) *D*ecide how to deal with ethical issues; and (6) *E*valuate, interpret and present the data. More specifically, the common procedure for user studies involving IR systems (also referred to as *study protocol* [Kel09] includes the following [HY07]:

1. Assign participants various 'realistic' tasks to perform.
2. Take quantitative measurements of 'performance' (e.g. time taken, number of tasks completed, number of errors made, etc.).
3. Make observations about how the interface/system is being used by the participants.
4. Collect subjective reactions from the participants (e.g. satisfaction, usability)

This experiment process requires careful design choices of several factors that can influence the results and reliability of the evaluation. The rest of this section discusses some

of these factors, including the experiment type (laboratory/operational), the experiment design (within/between- subjects), recruitment of subjects (e.g. selection procedure and number of subjects) and search tasks (e.g. number and type of tasks).

4.4.2.1 Lab-based versus Naturalistic Settings

Experiments can be carried out in a laboratory (also referred to as *controlled or formal experimentation*) or an operational (also referred to as *contextual inquiry* or *naturalistic observation*) setting. In the first setting, the experiment takes place in a laboratory where the researcher has (some) control over the experimental variables and environment. In the latter, the experiment takes place in the real operational/natural setting where the IR system is typically used. There are advantages and disadvantages acknowledged for each setting, and its choice should be carefully considered by the researcher [Ts92, Rob81, BI97, Pet08]. First, the laboratory setting offers more control of independent variables that may affect the outcomes of the experiment. This is difficult if not impossible to achieve in a real setting. This control can be a necessity in certain studies, particularly those attempting to answer specific questions or examine the effects of one or more variables, as opposed to open/free studies such as those analysing users behaviour or strategies during search. In these studies, the operational setting might be preferred since it provides the ability to observe users in real scenarios as opposed to simulating them. However, this realism is also a drawback of this setting, since, besides the lack of control mentioned earlier, it prevents the repeatability of the experiment, an important aspect especially in large-scale evaluations. In an attempt to have the best of both worlds, experiments can be performed in a combination of both settings [WJD90, WHB91, Rob81].

4.4.2.2 Within or Between Subjects Design

Another important choice for the experiment is with respect to the use of subjects. For example, if comparing two IR systems, subjects might be asked to test only one system (known as *between-subjects* or *independent* design) or test both systems (known as *within-subjects* or *repeated measures* design) [Kel09]. The advantages of a within-subjects design are that the subjects test all IR systems and therefore they are able to compare the results of multiple systems. In addition, fewer subjects are required to conduct the experiment because subjects test multiple systems. However, a disadvantage of this experimental design is the *carryover/order* effect. This occurs when the subject 'carries over' certain effects from the experiment with one system to the next system used. These effects include learning effects and pre-conditioning, as well as emotional effects, such as tiredness, boredom and frustration. Various techniques, such as *randomisation*, *rotation* and *counterbalancing* of search systems and topics have been used to overcome these effects [Ts92]. To apply counterbalancing, a *Latin Square* (to control effect of one variable) or *Graeco–Latin Square* design (to control the effects of multiple variables) are the most common approaches used for overcoming any topic or system ordering effects that may influence the results obtained [Kel09, pp. 44-60].

4.4.2.3 Recruitment of Subjects

Identifying the 'right' number of subjects to recruit for an IIR or usability study is an open question. [Nie93, Nie94, Lew94] and [Vir90, Vir92] argue that five or fewer subjects are sufficient to identify most of the usability problems found in a system. [Vir90, Vir92] show that this could be as many as 80% of the problems. However, other studies recommended using more subjects. For example, [TLN06] suggested that seven subjects may be optimal; [PL01] argued that more than eight subjects are required; [Spy92] called for using a minimum of 10-12 subjects and [Fau03] explained that 15 subjects are required in order to detect between 90 to 97% of the problems.

Similarly, this has been a difficult choice for IIR evaluations since it usually includes a trade-off between available resources and the reliability of the evaluations. The number of users directly influences the amount of resources required in terms of cost, time and effort, which are always limited in research studies. Additionally, there is the common difficulty of finding volunteers with the required characteristics such as a specific age group or knowledge in a particular field. On the other hand, insufficient numbers of subjects can directly affect the reliability of the results and their statistical significance, which risks the overall validity of the evaluation. For example, interactive CLEF [GO04, GCV06] used a minimum of eight subjects while the Interactive Track at TREC-9 and TREC-6 used 16 and 20 searchers respectively [HO99, Ove97].

Another important aspect with recruiting subjects is the choice of a representative sample of the target population. For instance, if an IR system is operationally used only by librarians then recruiting subjects from a different domain would bias the results. The type of users is also an important factor when recruiting. There have been two main approaches for categorising users: in terms of search experience and skills or in terms of domain/field experience and knowledge [HY01, HS00]. For instance, [NPSR99, HS00] and [TS05] differentiated between *inexperienced/casual users* and *experienced/expert users* according to their web expertise, which was defined by [HS00] as "the knowledge and skills necessary to utilise the World Wide Web and other Internet resources successfully to solve information problems". Evaluating systems with the different types of users and comparing their results could help in understanding the suitability of certain search approaches to specific types of users, and the different requirements to cater for when targeting one of these types of users. For instance, this approach was followed in TREC-6 Interactive Track [Ove97] in which the participating systems were evaluated by 8 librarians and 12 general users.

4.4.2.4 Tasks and Topics

Subjects may be provided with a set of tasks to carry out during an experiment, or may be asked to think of their own. Tasks could include: *specific fact-finding* (e.g. finding someone's phone number), *extended fact-finding* (e.g. finding books by the same author), *open-ended browsing* (e.g. identifying any new work on voice recognition from Japan), and *exploration* (e.g. finding out about your family history from an online archive) [Shn86, p. 512]. For more browsing-oritented tasks, such as exploration, then evaluation

may go beyond simply dealing with ranked lists of results (e.g. visualisations).

In an attempt to provide a balance between realism and control, [BI97] proposed the *simulated work task*, a short cover story describing a situation that leads to the information need. The authors explain how this helps in simulating real life by allowing individual interpretations of the situation. Typically a simulated work task situation includes: a) the source of the information need; b) the environment of the situation; and c) the problem which has to be solved. Researchers use one or more of these types of tasks depending on the research goal/questions. Search tasks/topics are used at TREC and CLEF interactive tracks [Ove97, GCV06] while [JFH98, Bor00, WBC07] and [Pet08] adopt Borlund's simulated work task. In the same context, another approach to achieve realism and motivate and engage the recruited subjects in the evaluation is to let them choose the search tasks from a pool of available tasks; for instance, in different domains [Spi02, Su03, WBC07, JHJ08].

Additionally, the number of tasks included in a user study is an important aspect to consider as this has an impact on cost, time and effort required to conduct the study. Some studies have imposed time limits on each task to allow for a specific number of tasks within particular period of time [Ove97, KA08]. For instance, six tasks were included in the TREC-6 Interactive Track with a time limit of 20 minutes for each task [Ove97]. The total amount of time required for the experiment was around three hours. This was the same amount of time which the organisers of iCLEF 2004 found to be the longest for subjects to offer in a single day [GO04]. However, in this study, 16 different tasks were performed by the subjects, far more than the small number of tasks commonly adopted in other studies [Bor00, Ove97, Kau07]. Indeed, the total amount of time required depends on many factors, including the type of tasks as well as steps performed in the experiment (e.g. briefing, questionnaires, interviews, etc).

4.4.3 Data Collection Methods

Another important design choice within IIR studies is with respect to the methods employed to collect the data generated from the experiments, which are influenced by the evaluation criteria and measures. For instance, to assess systems with respect to users' satisfaction, approaches for gathering such subjective data are used, such as questionnaires or interviews with the recruited subjects. In contrast, system processing logs are often used for collecting objective data such as different timings required for assessing user-efficiency. The rest of this section discusses some of the most common data collection methods used, together with their most important advantages or limitations as acknowledged in the literature.

4.4.3.1 Logs

Logs are also referred to as system processing logs, system usage logs or transaction logs. They are known as a type of *unobtrusive methods* which typically collect data without direct engagement of the user (such as observations by the experiment leader) [McG95, Pag00, WCSS00]. Although the main definition in literature for logs is a method that

captures the interaction between an IR system and its users including the content, type and time of transactions [RB83, Pet93], it is also used in IIR user-studies to refer to other types of data automatically recorded while users are performing the search tasks. This data is usually gathered for performance- or efficiency- based measures. An example is using software to record different timings such as search time per task or response time. Logs are widely used as an inexpensive data collection method – in contrast to questionnaires or interviews which require time from the recruited subjects – that allows the generation of large amounts of different types of data as required. For instance, this includes gathering the different number of search reformulations attempted by users for a search task or the time it takes them to formulate their queries in a specific interface.

4.4.3.2 Think Aloud

Think aloud [Nie93] is a well-established method for gathering data in user-studies [ES93, Ove01, JCW03, WKG+12]. In essence, it is based on attempting to gain more insight and understanding of the users actions and strategies, their interactions with and perception of the system, as well as their behaviour and rationale during different scenarios or search tasks. The users are therefore asked to *think-aloud* while performing the tasks, explaining what they are doing and why, and voicing any problems or difficulties they face. Various researchers showed that this method allows generating valid qualitative data that could be used to study cognitive tasks [RD90, ES93]. However, this validity was questioned by other researchers arguing that thinking aloud could influence users performance and behaviour during completing the tasks. For instance, [RD90] and [Bai79] found that thinking aloud slows down users in the main task. Additionally, [Ing92] argued against the reliability of the data generated from this method since there is no guarantee that it reflects the real behaviour of the users. Finally, since these loud-thoughts are usually audio-recorded, this results in the method being expensive, requiring large amount of resources (time, effort, personnel) to analyse the data.

4.4.3.3 Questionnaires

This is one of the most commonly used data collection methods in user-studies [Har96, Kel09]. Pre-search questionnaires are often used to gather information about subjects' knowledge, experience or skills in a specific field. Hence, they are common in studies investigating the effect of specific tasks or systems on subjects and the resulting changes in this knowledge. When included, the demographics questionnaire is used to gather subjects' demographics data. Most of the times, these two questionnaires are merged in one questionnaire presented to the subjects at the beginning of the experiment. The data collected can be used to identify correlations (for instance, between performance and age) or specific behaviour (for instance, different search strategies depending on level of experience). Typically, standardised questionnaires are utilised in post-task or post-experiment questionnaires aimed at capturing feedback about the usability of the system and subjects' satisfaction. System Usability Scale (SUS) [Bro96], Computer System Usability Questionnaire (CSUQ) [Lew95] and Questionnaire for User Interface

Satisfaction (QUIS) [CDN87] are some of the widely used satisfaction questionnaires in HCI. For instance, SUS comprises ten normalised questions which are answered on a 5-point Likert scale identifying the subjects' view and opinion of the system. The test incorporates a diversity of usability aspects, such as the need for support, training and complexity.

In addition to the above methods, most researchers also use observations (also referred to hidden information) to gain insight into the subjects' behaviour while performing the required tasks as well as any problems or difficulties facing them. Furthermore, interviews are also used as an alternative to or together with questionnaires to gather information about subjects' satisfaction.

4.5 Evaluation Initiatives

In 1992, the National Institute of Science and Technology (NIST) announced the first Text REtrieval Conference (TREC) to encourage IR community researchers to test their own algorithms and systems using a 'standard' test collection (1 million documents) and organised a meeting to compare and discuss the results. TREC continues to run an annual cycle of releasing re-useable benchmarks and meetings to discuss results – a process which has proven highly influential in the development of IR techniques [SJ00]. While 25 groups participated in TREC-1, the current number of participants is significantly higher (as too are the sizes of the datasets). Driven partly by the success and achievements of TREC, a number of other large-scale IR evaluations have been run. These include the Cross-Language Evaluation Forum (CLEF; started in 2000) for evaluating multimodal and multilingual information access methods and systems as well as evaluations in Interactive Information Retrieval (IIR), such as the ones embodied within TREC – *Interactive Track* [HO00] and *Complex Interactive Question-Answering* (ciQA) [KL07] – which involve real users to creating topics or evaluating documents. Another well-established evaluation series is the one organised by the INitiative for the Evaluation of XML retrieval (INEX) for evaluating XML retrieval methods and systems, with the first run in 2002 [FGKL02]. In the database community, the Wisconsin Benchmark [BDT83] was the first attempt to compare database systems developed for evaluating specific features [Bit85, SFGM93, Cat94].

Ontologies were at the forefront of early Semantic Web research and, as their number increased, the need for ontology matching evaluations became apparent. The *Ontology Alignment Evaluation Initiative (OAEI)* was founded in 2005 after merging two separate evaluation events[6] and has been driving progress in the area ever since. The evaluation of RDF stores also started around the same time (e.g. Lehigh University Benchmark [GPH05], Berlin SPARQL Benchmark [BS09], SP2Bench [SHLP08]).

Until recently, attempts to evaluate Semantic Search technologies were limited to isolated evaluations of newly developed approaches [BCC+08, LPM05] or comparing

[6]The Information Interpretation and Integration Conference (I^3CON) (http://www.atl.external.lmco.com/projects/ontology/i3con.html) and the EON Ontology Alignment Contest (http://oaei.ontologymatching.org/2004/Contest/)

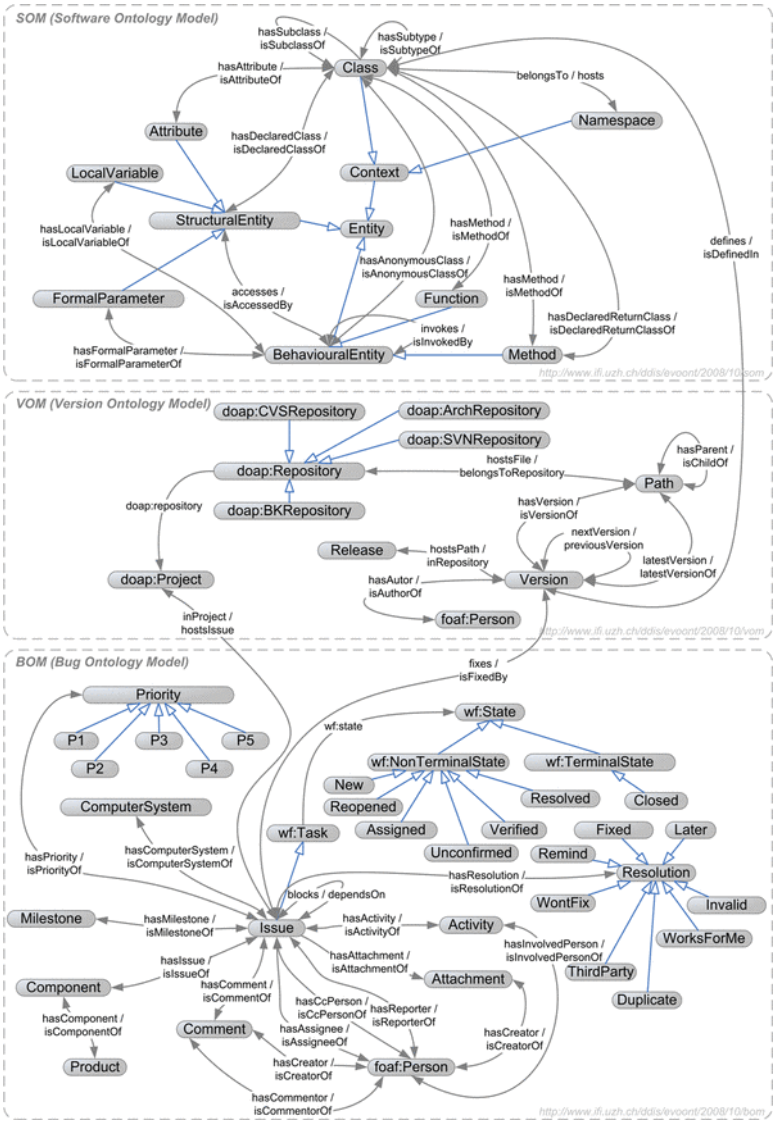


Figure 4.1: The three ontology models of EvoOnt from [TKB10].

the usability of different interfaces [Kau07]. Due to the lack of standardised evaluation approaches and measures, researchers applied a diverse set of both datasets and tasks and usually refrained from comparing their tools with other similar ones in the domain. Fortunately, the community recognised this lack of (and subsequent need for) a comprehensive evaluation to foster research and development. In the last three years, four different evaluation series were initiated, namely the SEALS semantic search evaluations [WRE+10, WGCT11], the SemSearch challenge [HHM+10], the QALD open challenge [UCLM11] and, finally, the TREC Entity List Completion task [BSdV10]. The remainder of this section describes each of these evaluation initiatives.

4.5.1 Semantic Evaluation at Large Scale (SEALS) - Search Theme

According to the organisers of SEALS semantic search evaluations, the group of tools considered are user-centred tools (i.e., are intended to interact with people not com-

puters) for retrieving information and knowledge [WRE+10, WGCT11]. This excludes tools which require structured queries as input as well as document retrieval systems which return results as Semantic Web documents relevant to the given query.

The methodology adopted in running the two evaluation campaigns – which took place in 2010 and 2012 – consisted of two phases: an *Automated Phase* and a *User-in-the-loop Phase*. These phases allowed tools to be evaluated in terms of both performance as well as usability and user satisfaction. Besides the performance, another criterion assessed in the automated phase was scalability: the ability of tools to scale over large datasets. In order to assess scalability, one or more datasets of different sizes were required. Therefore, the EvoOnt[7] software-engineering dataset was chosen by the organisers for this phase. Gradient ABox sizes ranging from 1k to 10M triples were created with the same TBox. A graphical representation of the EvoOnt ontology is presented in Figure 4.1.

Additionally, queries used in this phase were based on templates created after conducting experiments with professional programmers for the identification of standard and useful software engineering questions that programmers tend to ask when evolving a code base [dAM08, SMDV06, SMV08]. The 50 queries were generated to include ones with varying levels of complexity: simple ones such as *'Which **methods** have the **declared return class** x?'* and more complex ones such as *'Give me all the **issues** that were reported by the **user** x and have the **state** fixed.'*[8]. The groundtruth for the final set of queries were generated by running the SPARQL query equivalent to the NL one on the dataset. Similar to IR evaluations, precision, recall and f-measure were computed as well as other performance measures such as the execution time (speed), CPU load and amount of memory required.

In the user-in-the-loop phase, the geography dataset (shown in Figure 4.2) from the Mooney NL Learning Data[9] was selected since the domain is sufficiently simple and understandable for non-expert end-users. NL questions for this dataset were already available and therefore were used as templates to generate queries for the evaluation for which the groundtruth was also available with the question-set. Again, questions were chosen to range from simple to complex ones as well as to test tools' ability in supporting specific features such as comparison or negation. Simple questions included ones such as *'Give me all the **capitals** of the* USA*?'*; more complex questions included ones such as *'What are the **cities** in **states** through which the* Mississippi *runs?'*; and, finally, questions such as *'Which **lakes** are in the **state** with the highest point?'*[10] tested the ability for supporting superlatives (highest point).

The two usability experiments were conducted in a laboratory setting in which the recruited subjects were given a number of questions to solve with one or more tools. The first evaluation campaign (2010) used a between-subjects design in which each participating tool was evaluated with 10 different users. Although this experiment yielded a useful set of findings and recommendations for the community, it did not allow di-

[7] https://files.ifi.uzh.ch/ddis/oldweb/ddis/research/evoont/index.html
[8] Concepts are shown in bold
[9] http://www.cs.utexas.edu/users/ml/nldata.html
[10] Concepts are shown in bold

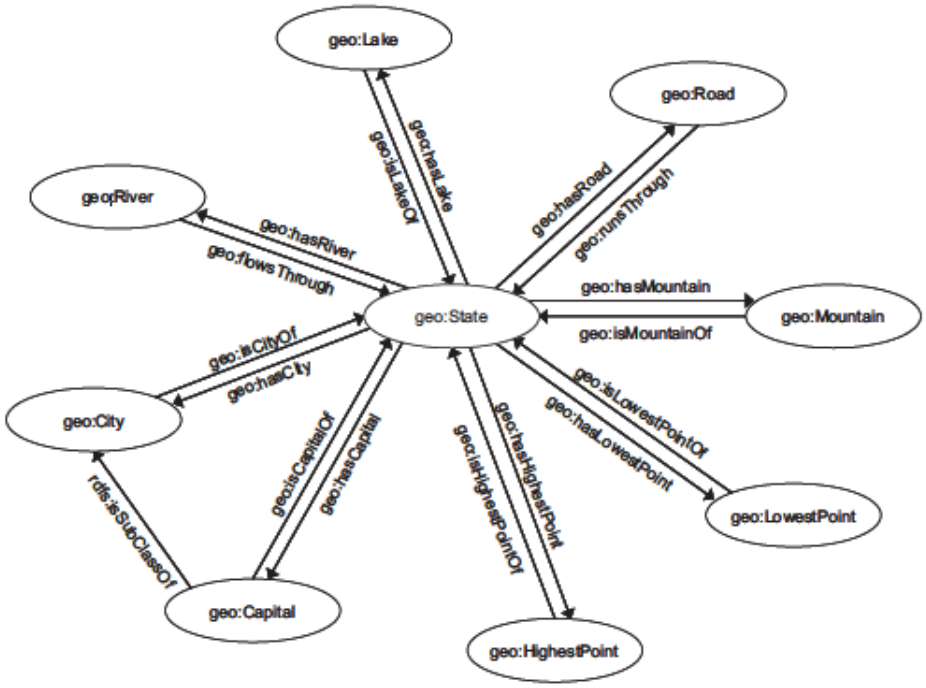


Figure 4.2: The ontology model of the geography dataset in Mooney from [Kau07].

rect comparison of the different tools and their employed query approaches. This was addressed in the second evaluation campaign (2012), where tools were evaluated using a within-subjects design setting with 10 casual users and 10 expert users. Here, the subjects evaluated the five tools in randomised order to avoid any learning, tiredness or frustration effects that could influence the experiment results.

For each tool, subjects were given a short training session explaining how to use the tool to formulate queries and become familiar with the evaluation control software (used for issuing new questions and collecting feedback). Following the training period, subjects were asked to formulate each question using the tool's interface. The order of the questions was randomised to avoid any learning effects. In both evaluations, the objective data collected included: 1) input time required by users to formulate their queries, 2) number of attempts, and 3) answer found rate capturing the distinction between finding the appropriate answer and the user 'giving up' after a number of attempts. Additionally, subjective data was collected using two post-search questionnaires to capture users experience and satisfaction. In the second evaluation, to allow direct comparison, users were asked to explicitly rank the tools according to certain criteria such as how much they liked the tools or how much they found the results to be informative and sufficient.

4.5.2 SemSearch

[PMZ10] explain how object-retrieval can be seen on the Web of Data as the counterpart of document retrieval on the Web of Documents, since the first is about resources that represent objects and the latter is about resources that represent documents. They

define *object retrieval* as 'the retrieval of objects in response to user formulated keyword queries' and present a classification for these queries according to the primary intent of each query. The most common type is the *entity query* which requires information about a specific instance on the Web of Data (e.g. 'IBM'), followed by the *type query* which asks for instances of a given type (e.g. 'films for Julia Roberts'). The least common types are the *attribute query*, which requires information about an attribute of a type or an entity, and the *relation query*, in which information about a specific relation between types or entities is requested. If the query does not fall in any of the previous types, it is then classified as *other keyword query*.

Recognising object retrieval as an integral part of semantic search and following the methodology defined by [PMZ10], [HHM+10] organised a challenge which ran twice within the SemSearch workshops (in 2010[11] and 2011[12]) with a focus on *ad-hoc object retrieval*. The input query is given as a set of keywords and the expected output is a ranked list of object URIs retrieved from an RDF dataset. The requirements for the dataset were: 1) to contain and thus represent real data found on the Semantic Web, 2) to be of large yet manageable size and 3) not biased towards one particular semantic search system. Therefore, the 'Billion Triples Challenge 2009 (BTC-2009)' dataset was chosen. It contained 1.4B triples with data about 114 million objects, crawled by multiple semantic search engines: FalconS [CWGQ08], Sindice [TOD07], Swoogle [DFJ+04], SWSE [HHD+07], and Watson [dBG+07], during February/March 2009.

According to the previous query classification, the 2010 evaluation focused on *entity queries*, while *type queries* were added in the 2011 evaluation. As with the dataset selection, the query requirements were: 1) to represent real-world queries given by actual users, and 2) to be unbiased towards one specific semantic search system. To conform with these requirements, the organisers decided to use queries from logs of traditional search engines as opposed to ones from semantic search engines. They argued that the latter – largely centred around *testing and research* – did not represent real information needs of 'casual users' (at least not at the time of the evaluations). In the 2010 evaluation, the *entity queries* used were selected from the logs of Yahoo! Search and Microsoft Live Search and included ones such as 'Scott County', *'american embassy nairobi'*, *'ben franklin'* and *'carolina'*. This was, however, different in the 2011 evaluation in which the *type queries* were 'hand-written' by the organising committee and included ones such as *'Apollo astronauts who walked on the Moon'*, *'movies starring rafael rosell'*, and *'wonders of the ancient world'*.

Only the first 10 results per query were considered, each of which was assessed on a 4-point scale of relevance (0 being not relevant to 3 being a perfect match). The assessment was carried out by human judges using Amazon's Mechanical Turk crowdsourcing platform. The measures used to evaluate and compare the performance of the participating tools were the *normalised Discounted Cumulative Gain (nDCG)*, *Mean Average Precision (MAP)*, and *Precision at rank k (P@k)*. After the 2010 evaluation, it

[11] http://km.aifb.kit.edu/ws/semsearch10/
[12] http://km.aifb.kit.edu/ws/semsearch11

was observed that most of the participating systems used IR-based techniques and made little use of the semantic data for more advanced reasoning and retrieval. Additionally, the systems did not try to use semantic-based techniques for understanding or expanding the queries.

4.5.3 Question Answering Over Linked Data (QALD)

As opposed to keywords-based search interfaces, question answering systems allow users to express more complex information needs. The Semantic Web community has significant research related to developing natural language-based interfaces for closed domain RDF data. However, there has been little progress in scaling these approaches to deal with linked data with its heterogeneous, noisy, distributed and open nature. The QALD open challenge, with the aim of advancing this topic and facilitating the evaluation of such approaches, focused on evaluating question-answering systems that help users find answers for their information needs in semantically annotated data using a natural language interface.

The challenge has taken place three times (in 2011[13], 2012[14] and 2013[15]) with two different tasks: an open/cross-domain task and a closed-domain one. It is interesting to note the need raised by the organisers for facilitating multilingual access to semantic data by changing the third challenge (2013) to be on multilingual question answering over linked data.

DBpedia was chosen for the first task since it allows testing the ability of the participating systems to scale over large datasets commonly found in the Semantic Web. Additionally, since DBpedia is an extraction of structured data from Wikipedia, it allows testing the systems' ability to deal with noisy data featuring various levels of quality. Moreover, it contains data spanning multiple domains and thus conforms with the heterogeneity requirement. DBpedia 3.6, 3.7 and 3.8 were used in the 2011, 2012 and 2013 evaluations, respectively. The first contained around 670 million triple, the second consisted of 1 billion triples, while the third consisted of 1.89 billion triples. Additionally, in the third evaluation (2013) and to facilitate the multilingual task, DBpedia 3.8 was provided with multilingual labels, in addition to the Spanish DBpedia[16].

In the closed-domain task, an RDF export of MusicBrainz[17] – an open music encyclopedia that collects music metadata and makes it available to the public – was used. Its RDF export contains a small ontology describing the music domain and comprises only a few classes and relations as opposed to DBpedia whose ontology contains more than 320 classes. There are approximately 25 million triples describing artists, albums, and tracks, as well as a subset of important relations between them. Both datasets were selected to represent data found in the Semantic Web since they have been widely used in the research community and largely interlinked with other datasets in the LOD

[13] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge&q=1
[14] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge&q=2
[15] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=home&q=3
[16] http://es.dbpedia.org/
[17] http://musicbrainz.org/

Table 4.1: Semantic Search Evaluations

Evaluation Name	**Dataset**	**# of Queries**	**Query Type**	**# of Participants**
SEALS-1 automated	EvoOnt	50	Artificial	4
SEALS-1 uitl	Mooney-Geography	20	Artificial	4
SEALS-2 automated	EvoOnt	50	Artificial	5
SEALS-2 uitl	Mooney-Geography	5	Artificial	5
SemSearch-1	BTC 2009	92	Real-world	6
SemSearch-2	BTC 2009	50 (entity), 50 (list)	Real-world	5
QALD-1 closed-domain	MusicBrainz	50	Artificial	2
QALD-1 open-domain	DBpedia 3.6	50	Artificial	2
QALD-2 closed-domain	MusicBrainz	55	Artificial	0
QALD-2 open-domain	DBpedia 3.7	100	Artificial	4
QALD-3 closed-domain	MusicBrainz	100	Artificial	1
QALD-3 open-domain	DBpedia 3.8	100	Artificial	6
TREC ELC-1	BTC 2009	8	Artificial	3
TREC ELC-2	Sindice-2011	50	Artificial	7

cloud[18].

In the three evaluations, a training set of natural language questions together with their equivalent SPARQL queries and correct answers were provided for each task prior to the challenge. Another set of questions were used to evaluate and compare the participating systems with respect to precision and recall. For the multilingual task in the third evaluation (2013), all questions were provided in six languages: English, Spanish, German, Italian, French, and Dutch.

The training and test questions were written by students who explored the dataset in an attempt to simulate real users' information needs. Also, the questions were not biased towards one specific approach. For instance, questions used in the open-domain task included ones such as *'which river does the Brooklyn Bridge cross?'* and *'give me all cities in New Jersey with more than 100,000 inhabitants'*. The closed-domain task included questions such as *'give me all soundtracks composed by John Williams'* and *'which bands released more than 100 singles?'*. Finally, it is not clear how the groundtruth for the questions were generated, but it is highly possible that they were manually produced by the evaluation organisers.

[18] http://lod-cloud.net/

4.5.4 TREC Entity List Completion (ELC) Task

Similar to SemSearch, the importance of entity-oriented search to address different information needs concerning entities on the Web was recognised by the TREC community. [BMdR10] categorises different tasks of entity-oriented search as follows:

1. *entity ranking*: given a query and target category, return a ranked list of relevant entities.
2. *list completion*: given a query and example entities, return similar entities.
3. *related entity finding*: given a source entity, a relation and a target type, identify target entities that exhibit the specified relation with the source entity and that satisfy the target type constraint.

The second task (*list completion*) is the focus of the *Entity List Completion (ELC)* task found in TREC Entity Track. It is similar to the SemSearch *type queries* in that both limit their queries to ones that require instances of a specific type. However, the ELC task is more specific since each query requests instances of a specific type that are related to a given entity with a given relation.

Again, the BTC-2009 dataset used in SemSearch was used in the first year of running the ELC task. In the second year the organisers used Sindice-2011, a more entity-oriented dataset which is *especially designed for supporting research in the domain of web entity retrieval* [CCP+11]. Queries were selected from the REF-2009 topics[19] according to their suitability to the task and the dataset: for example, having information about the query entities in the dataset. Additionally, each query included a considerable amount of information for the participating groups, such as the URI of the given entity on the Web of Data, a DBpedia class representing the target entity type as well as URIs for examples of the expected instances. A query example is given below:

```
<query>
<num>7</num>
<entity_name>Boeing 747</entity_name>
<entity_URL>clueweb09-en0005-75-02292</entity_URL>
<target_entity>organisation</target_entity>
<narrative>Airlines that currently use Boeing 747 planes.
</narrative>
<entity_URIs>
<URI>http://dbpedia.org/resource/Boeing_747</URI>
</entity_URIs>
<target_type_dbpedia>dbpedia-owl:Airline</target_type_dbpedia>
<examples>
<entity>
<URI>http://dbpedia.org/resource/Northwest_Airlines</URI>
<URI>http://www.daml.org/2002/08/nasdaq/nasdaq#NWAC</URI>
</entity>
<entity>
<URI>http://dbpedia.org/resource/British_Airways</URI>
<URI>http://twitter.com/British_Airways</URI>
</entity>
```

[19] http://ilps.science.uva.nl/trec-entity/guidelines/guidelines-2009/

```
...
</examples>
</query>
```

Types found in the topics (e.g. airlines) were mapped to the most specific class within the DBpedia ontology (e.g. dbpedia-owl:Airline). Results returned by participating systems were requested to be in the form of a ranked list of URIs which were assessed on a binary relevance scale (relevant or irrelevant). Only the first 100 results were considered and judged by the staff at NIST and by the track organisers. The evaluation measures used in the first run (pilot) were the Mean Average Precision (MAP) and R-Precision. For the second run, the normalised Discounted Cumulative Gain (nDCG) was the main measure used.

4.6 Summary

This chapter has reviewed the literature on evaluations and their design in IR, in addition to existing semantic search evaluations. Since a main part of the thesis is conducting evaluations of (semantic) search systems, this chapter provides the required background and knowledge required to conduct such evaluations. The review of evaluations in IR is included with the intention of learning from this community, as it has been an active area of research for the past 50 years and the subject of much discussion and debate. Firstly, the two evaluation paradigms in IR, the system-oriented and user-oriented approaches, have been discussed, and various aspects relevant to each of them have been presented. For instance, the system-oriented approach has been focusing on assessing retrieval performance, as opposed to users' satisfaction and experience. Therefore, in the system-oriented approach aspects such as selection of the document collections and queries, the judgment process and evaluation measures have been discussed. Related to the the user-oriented approach, aspects such as the different criteria to be assessed, the experiment setup and the data analysis and collection methods have been reviewed. Furthermore, current semantic search evaluation initiatives including the SemSearch Challenge, the SEALS semantic search evaluations, the QALD open challenge and the TREC Entity List Completion task have been reviewed with respect to the most relevant aspects from the ones previously mentioned.

Chapter 5

Analysis of Semantic Search Evaluation Initiatives

This chapter provides a thorough analysis of the semantic search evaluations reviewed earlier in Chapter 4 highlighting the limitations and deficiencies in each of them. In the rest of the chapter, the four reviewed evaluations are analysed with respect to the core aspects necessary to conduct an effective evaluation (see Section 4.2): datasets, queries, relevance and judgments, and measures. Note that aspects related to user-based evaluations are not analysed here since none of the evaluations included this scenario except SEALS, and a summary of the latter has been provided in Section 4.5.1. Indeed, the main motivation behind this analysis is to understand whether any of these current evaluations is suitable for answering the required research questions (presented in Chapter 6); that is, to investigate the usability and learnability of different semantic search query approaches.

5.1 Datasets

5.1.1 Origin

As discussed in Section 4.3.2, evaluation datasets are either specially-created or operationally-derived. Of all the datasets used (see Table 5.1) in the evaluations described in Section 4.5, three were specially-created: EvoOnt, Mooney and Sindice-2011. The EvoOnt dataset was chosen for the automated phase in SEALS since it provided the ability to assess the *scalability* criterion by creating datasets of various sizes given the same ontology. The Mooney dataset was chosen for the SEALS user-in-the-loop phase since it described a simple and common domain which allowed easily understandable questions. Finally, Sindice-2011 was used in the TREC ELC task evaluation since it provided an entity-oriented dataset, specifically designed for supporting research in web entity search.

The core benefit of using a bespoke dataset is the control it allows over certain features of the dataset (see Section 4.3.2) and, as a result, other aspects of the evaluation:

Table 5.1: Properties/features of the datasets used in the reviewed evaluations.

Dataset	**Type/Nature**	**Domain**	**Size (triples)**	**Creation Year**
Mooney	Specially-created	Closed: Geography	5700	2001
EvoOnt	Specially-created	Closed: Software Engineering	Not fixed	2007
DBpedia	Operationally-derived	Heterogenous	1 billion	2009
MusicBrainz	Operationally-derived	Closed: Music	25 million	2007
BTC-2009	Operationally-derived	Heterogeneous	1 billion	2009
Sindice-2011	Specially-created	Heterogeneous	11 billion	2011

the task to be evaluated (e.g. entity-retrieval), the type of the evaluation (e.g. usability), or the assessed criteria (e.g. scalability). However, this level of control comes at the cost of representativeness:

- In principle, the evaluated systems are going to be used in the real world and thus should be assessed for their ability to work with real data.
- It is difficult to simulate some of the aspects found in real data such as the various levels of quality or the noise and errors typically found in it.
- A specially-created dataset is usually designed with specific characteristics or to test specific features defined by the evaluation organisers or by domain experts. However, there is no real guarantee that these are the right/appropriate characteristics of real data, which again raises the question of representativeness and realism.

One could argue that the Sindice-2011 dataset is operationally-derived since it is based on real data crawled by Sindice. However, the counterargument is that the original Sindice crawl was not provided *as-is* but was processed and transformed into a corpus of entities. This processing of the data invalidates, to a certain degree, the realism aspect. To illustrate, the authors mention that "*we filter all entity graphs that are composed of only one or two triples. In general, such entity graphs do not bear any valuable information, and can be removed in order to reduce the noise as well as the size of the dataset*" [CCP+11, p. 28].

It is important to note that even *operationally-derived* datasets may be subject to some degree of 'cleansing' before release, thus reducing their representativeness. Furthermore, DBpedia — structured data extracted from Wikipedia — is not as diverse or noisy as the BTC-2009 dataset, which is based on data crawled by Falcon-S, Sindice, Swoogle, SWSE and Watson.

If possible, datasets ought to be *operationally-derived*. Despite the choice of EvoOnt by the SEALS initiative being due to the apparent ease of creating multiple datasets of differing sizes [BRWC09], one could imagine using the same tools to create test data of various sizes for the same ontology but for an operationally-derived dataset (e.g.

DBpedia). The ability to create such datasets (common ontology, varying size) which are free from inconsistencies would be beneficial to the community and ought to be investigated. Similarly, *operationally-derived* datasets covering easily understandable domains — c.f., SEALS' choice of Mooney for their usability study [BRWC09] — are available: Geonames[1] in the geography domain, MusicBrainz[2] in the music domain or DBpedia in both as well as other domains.

5.1.2 Domain

Of the datasets shown in table 5.1, an equal split between *closed-domain* and *open-domain* can be observed. While Mooney, EvoOnt and MusicBrainz are closed-domain datasets (describing a single domain such as Geography), DBpedia, BTC-2009 and Sindice-2011 are heterogeneous datasets spanning multiple domains. Indeed, *open-domain* data can be argued to be increasingly important (at the expense of *closed-domain* data); the size of (open) linked data is continuously increasing and offers significant potential for answering various information needs. In response, semantic search development has begun to focus on search of the open web as opposed to traditional, closed-domain approaches. However, with the proliferation of heterogeneous datasets, more care than ever must be taken when choosing which datasets are selected to evaluate systems — the datasets must be applicable to the system and task and representative of the types of data for which the tool is designed.

5.1.3 Size

Dataset size has a strong influence on the evaluation criteria and the subsequent reliability of the evaluation and its results. In IR, the definition of a *large* dataset (i.e., sufficient for running realistic evaluations) has not been fixed and is continuously growing to reflect the increasing amount of data commonly available to organisations. For instance, when Sparck Jones and van Rijsbergen described *ideal test collections* in the 1970s, they were referring to datasets containing around 30,000 documents [SJVR76]; current TREC test collections can contain a billion documents[3]. Open-domain datasets used in semantic search evaluations should reflect the growth of the Semantic Web and linked data; with current datasets reaching billions of triples, the Sindice-2011 dataset comprising 11 billion triples could be considered to be more suitable for evaluating scalability than EvoOnt's 10 million triples.

Similarly, closed-domain evaluations should favour larger datasets. For instance, Geonames' 150 million triples is more representative than Mooney's 5000 triples. The argument for selecting Mooney for the SEALS evaluation's usability phase was its easily understandable domain. However, one could equally argue that this affects the reliability of the usability experiment since it should assess the user experience in a real-world scenario; given larger datasets covering similarly understandable domains (to non-experts

[1] http://www.geonames.org/ontology/documentation.html
[2] http://musicbrainz.org/
[3] http://plg.uwaterloo.ca/~trecweb/2012.html

in the case of SEALS), compelling arguments must be made for selecting a small dataset instead.

5.1.4 Age

The creation date of a dataset could also affect its suitability for a realistic and reliable evaluation. For instance, the BTC-2009 dataset was crawled during February and March 2009; naturally, this snapshot does not include any subsequent updates (such as DBpedia's September 2011 improvements to the schema, data and mappings). Given the speed with which the Semantic Web and linked data is evolving, preference should be given to newer datasets or datasets which receive regular updates.

5.2 Queries

5.2.1 Real Versus Artificial

Queries used in IR and similarly in semantic search evaluations are either *real queries* describing genuine information needs or *artificial queries* created to simulate the first. As shown in Table 4.1, SEALS, QALD and TREC adopted the second approach, in which queries are created either by domain experts, volunteers (usually students) or the evaluation organisers. In the SEALS user-in-the-loop phase, queries associated with the Mooney dataset were collected from university students as well as real users (*volunteers*), while in the automated phase queries were based on templates gathered from professional programmers (*domain experts*). In QALD, queries for both the closed- and open-domain tasks were written by university students who explored the MusicBrainz and DBpedia datasets; for TREC, topics from the traditional document-retrieval REF task were adapted to the ELC task.

There are arguments for and against the use of *artificial queries* over *real queries*. For instance, one could argue that gathering queries for a software engineering dataset from professional programmers is the nearest of all these ways to simulate real information needs since they are actual users in this domain. Similarly, university students could be seen as potential/real users for the music domain. However, although artificial queries allow for increased control and flexibility over the query content and features, the fact that they remain 'artificial' means they do not fully reflect real information needs. For instance, it was commonly reported in the SEALS usability experiments that the questions used were too difficult and would not be typed into a search engine by real users. An example is the question "*Which states have an area greater than 1 million and border states that border states with a mountain and a river?*". The argument in favour of these complex questions is again related to the evaluation task and expectations. To fully exercise the advanced semantic search systems, tasks ought to involve complex reasoning and integration of information to answer such questions. Thus, the real challenge is to produce a set of questions that can test this ability and other features of the systems while being more natural and representative of real information needs.

The approach of adapting topics from a traditional document-retrieval task (adopted in TREC ELC) was criticised by [PMZ10] who opted in SemSeach challenge for using *real queries* from query logs of *traditional* search engines. They argued the latter provided a better representation of real users' information needs than those from Semantic Web search engines. Additionally, such queries would not be biased towards any particular semantic search engine. We disagree with this argument; using logs of queries searching for documents on the Web for evaluating tools searching for knowledge on the Semantic Web is not an optimal choice. Queries posed to traditional search engines, such as Google and Yahoo!, are for different tasks and use cases and thus have different characteristics. Web users are not expecting actual answers to their queries; instead, they know they will obtain a list of documents that they can use as a starting point to navigate to other relevant documents that might contain answers to their information needs or investigate the top ones to extract the necessary answers. The difference in the nature of the task and format of results expected can change the way users formulate their queries; they are continuously learning to adapt their queries to the nature and capabilities of these search engines. For example, they learn how to overcome the limitation of handling complex queries that need integration of information by issuing multiple subqueries, investigating the results separately and manually forming the answers they are looking for. Therefore, although sampling the query logs of traditional search engines provides 'real' queries, each individual query does not necessarily capture the full complexity or richness of the original information need.

5.2.2 Query Set Size

A critical factor in the logistics of the evaluation execution, the analysis of the results, and ensuring the representativeness and coverage is the number of queries used. Table 4.1 shows that most of the reviewed evaluations used approximately the same number of queries (between 50 and 100). The first exception is regarding the number of questions used in the user-in-the-loop phase in SEALS during the two evaluation runs: 20 and 5, respectively. Indeed, for a usability study with real users, the number of questions should be carefully chosen to have reliable results without overwhelming the users. After the first evaluation run of SEALS, it was suggested that the use of 20 questions was too many and that, to keep the experiment within a reasonable duration and also to avoid subjects' tiredness or frustration, the number of questions ought to be reduced. The other exception is with the number of queries (only eight) used in TREC ELC-1. This was due to the adaptation of the queries from the REF task (see Section 4.5.4) to the ELC task, which proved to be problematic: queries were excluded since the dataset did not contain relevant entities for answering them, and thus only eight queries could be adapted.

For their proposed *ideal test collection*, Sparck Jones and van Rijsbergen suggested an acceptable number of around 250 queries, while 1000 queries might be needed in some scenarios. They claimed it was not useful to have less than 75 queries [JB77]. However, this was never achieved in IR, even with the huge increase in the size of document

Table 5.2: Semantic Search Evaluation Measures

Evaluation Name	**Relevancy scale**	**Judgments**	**Measures**
SEALS automated	binary	mechanised	Precision, Recall, F-Measure, Execution Time
SEALS uitl	binary	mechanised	Input Time, No. of attempts, No. of queries answered, SUS score, Extended score
SemSearch	three-point	manually (amazon mechanical turk)	MAP, P@k, NDCG
QALD	binary	mechanised	Precision, Recall
TREC ELC	binary	manually (track organisers)	MAP, R-Precision, NDCG

collection within earlier (e.g. Cranfield) and current evaluations (e.g. TREC). The common number of queries/topics currently used in TREC evaluations is 50[4]. This is due to the difficulty of obtaining a large number of topics and the cost involved in producing relevance judgements for each topic (see Section 4.3.4).

The number of queries to be used should be carefully considered within the evaluation design: it directly affects the stability of the evaluation measures and in turn the reliability of the evaluation results. For instance, [BV00] confirmed that results of evaluations using more topics are more reliable than those from evaluations using fewer topics. However, the exact number required differs according to the evaluation measure selected since they differ with respect to their stability. The authors showed that *precision at 10 (P@10)* has more than twice the error rate associated with it than the error rate associated with the *average precision*. In a later study, [VB02] found a strong, consistent relationship between the error rate, the size of the difference in scores, and the number of topics used. To conclude, this discussion suggests that while it might be more practical and pragmatic to use a small numbers of queries, it is essential to select the appropriate combination of evaluation measures and differences in scores (to differentiate between 2 or more methods) that corresponds nicely to the number of queries selected in order to achieve reliable results.

5.3 Relevance and Judgments

Recall from Sections 4.3.4 and 4.3.5 that different scales of relevance (such as binary, graded and ranked relevance) have been adopted in IR and semantic search evaluations. As shown in Table 5.2, a binary scale was used in SEALS, QALD and TREC evaluations, while a three-point graded scale was used in the SemSearch evaluation. As explained in Section 4.3.4, a binary scale is required for the use of precision and recall, the evaluation measures employed by SEALS and QALD. SemSearch organisers opted to use a three-

[4] http://plg.uwaterloo.ca/~trecweb/2012.html

point scale with *excellent* (result describes the query target specifically and exclusively), *not bad* (result mostly about the target) and *poor* (result not about the target, or mentions it only in passing). They found that, especially for expert judges, there is almost no difference between the two- and three-point scales and concluded that "*...there is thus no marked difference between a three-point scale and a binary scale*" [HHM+10].

As described earlier, the judgment process for the relevance of documents with respect to a given query is performed either automatically using a predefined groundtruth or manually by human judges. In the former, the groundtruth generation for each query is generally performed either by human judges scanning the entire document collection (or merely a sample), or by merging the results returned by different retrieval systems.

Neither approach was used in QALD and SEALS. Instead, it was generated by executing a SPARQL query equivalent to the NL query against the dataset and using the results as the groundtruth. This approach can be criticised: the transformation from natural language to SPARQL is a non-trivial task and errors can be introduced or indeed suboptimal SPARQL queries could be created. This is partly due to issues such as the large number of properties that can be found in the same dataset which refer to the same real-world relationship. This problem could, for instance, result in a SPARQL query that uses only a subset of these properties and misses some relevant results, leading to an incomplete result set affecting precision and recall.

TREC and SemSearch both used relevance judgments created by human judges. This approach is known to have a high overhead which increases in proportion to the number of results to be considered. Therefore, only the first 10 results are evaluated in SemSearch while this number increases to 100 in TREC (see Table 5.2). [PMZ10] argue that having more queries with fewer results evaluated (SemSearch: 92 queries and 10 results evaluated) is more desirable for web search engine evaluation than having few queries with more results evaluated (TREC: 8 queries and 100 results evaluated), especially when it is known that web users tend to examine only the top few results. Further work is required to establish an optimal tradeoff between these two factors to ensure reliable evaluations. Additionally, the other challenge facing this approach is the subjectivity of relevance judgments and the degree of inter-judge agreement that needs to be established to obtain reliable results. The judgments were performed by the track organisers in TREC and by Amazon's Mechanical Turkers in SemSearch. Although it is difficult to ensure the same level of subject knowledge for all judges, the deviation can be much greater with random volunteers than with evaluation organisers. It is indeed important to understand how this factor affects the reliability of an evaluation's results since it has been acknowledged in literature that the more knowledge and familiarity the judges have with the subject area, the less leniency they have for accepting documents as relevant [RS67, Cua67, Kat68]. Interestingly, [BHH+13] analysed the impact of this factor on the reliability of the SemSearch evaluations and concluded that 1) experts are more pessimistic in their scoring and accept less items as relevant when compared to workers (which agrees with the previous studies) and 2) crowdsourcing judgments cannot replace expert evaluations.

Relevance judgments are also affected by the amount of information provided as part

of the query itself. SemSearch used keyword queries such as 'american embassy nairobi' or 'ben franklin'. In contrast, TREC ELC used 'topics' which, as well as the entity name to search for, provided supplementary information such as the URI of the given entity on the Web of Data, a DBpedia class representing the target entity type as well as URIs for examples of the expected instances. We believe this is an important aspect in an evaluation design since the *amount of information* was found to be the highest-ranked factor affecting relevance judgments: "*such information would not only help one locate relevant results but also judge their relevance subsequently*" [Chu11, p. 271].

5.4 Measures

Both SEALS and QALD used set-based measures that do not take ranking into consideration (see Table 5.2). This is difficult to justify (barring grounds of simplicity) since one of the key features required from (semantic) search systems is to rank results according to relevance to a given query. Users will not examine hundreds, if not millions, of results even if they were actual answers rather than documents. In contrast, SemSearch and TREC used ranked-based measures. The first used precision@10, although as shown in Section 4.3.5, it is known to be the least stable among the other ranked-based measures (R-Precision, MAP, nDCG). Fortunately, it was used together with MAP and nDCG which provided both an overall figure of the systems' performance for the set of queries used (MAP) as well as their performance in retrieving highly relevant documents and ranking them (nDCG). nDCG, which is also adopted in TREC, has seen increasing use in evaluations based on non-binary scales of relevance. Despite being a well-established and reliable evaluation measure, using nDCG requires deciding on the values of the gain vector and the discount function, as well as producing an ideal ranked list of the results [Voo01, ZZXY08, KA09]. These details are unclear for both initiatives[5]; our best guess for the discount function is that both used the default one: log of the document's rank (see Section 4.3.5.2). It is worth noting that creating the ideal result list could be an even more challenging task than deciding these values, especially with the increase in the size of the datasets used, which in turn increases the difficulty of manually examining them to produce this list. Finally, R-Precision was the third measure used in TREC as a more stable and reliable measure than precision at a fixed level (P@k), which is used in SemSearch.

5.5 Summary

In this chapter, I have presented a thorough analysis of existing semantic search evaluation campaigns with respect to a number of critical aspects such as the datasets and queries used; the process of the result relevance decision; and the performance measures and how they are computed. Based upon this analysis, I have discussed limitations and flaws to the approaches followed in these evaluations.

[5]This is a long-standing criticism of such IR evaluations: critical details of the methods and measures adopted and the justification for them are described briefly or omitted. See Section 4.1

Recently, more attention is being given to evaluating semantic search tools, especially with the growth in development and research in this area. However, these efforts have largely been developed in isolation with no coherent overall design, leading to slow progress and low interest when compared to other established evaluation series such TREC in IR. This work is a first step towards identifying the adequacy and deficiencies of current evaluations as well as missing aspects in order to arrive at a more comprehensive and improved evaluation methodology and framework. This would enable more reliable and thorough assessments of semantic search tools and highlight their strengths and weaknesses, which would in turn drive progress and improvements in this area.

Part II

Methodology

"*He who performs not practical work nor makes experiments will never attain to the least degree of mastery.*"

– Jaber ibn Hayyan

Chapter 6

Requirements and Design: A User-Oriented Semantic Search Query Approach

As outlined in Chapter 1, the main research question this thesis attempts to answer is how to design a user-oriented semantic search query approach that is effective and usable beyond current state-of-the-art approaches. The thesis claims that to answer this question; a necessary step is to first evaluate the usability and effectiveness of current query approaches and the learnability of the best performing approach as perceived by the target users. Therefore, this chapter presents the requirements for such an effective and usable query approach, and for these user-based evaluations. Then, it discusses the design choices followed while answering the research question in order to address these requirements.

6.1 Requirements For A User-Oriented Semantic Search Query Approach

As stated throughout the thesis, the very broad goal of a (semantic) search system is to assist users in fulfilling their information needs. Although users' experience and satisfaction with this process is influenced by different aspects, including the effectiveness of the search system as well as the presentation of the results returned, the query format/approach is the starting point at which users can be guided to make their query in a way that will produce relevant results. The expressiveness of the query language together with the usability and the support provided by the query approach make a great deal of difference to whether users can successfully express their queries and find satisfactory answers. Therefore, the main focus of the work presented here, which will be shown in the requirements listed below, is on the query approach.

Most semantic search systems developed – up to the time of writing this – adopt a specific query approach that claim to be the best at tackling a specific challenge or

answering the needs of a specific type of users. Indeed, some of them assess the success of the system in doing the required task or the satisfaction of the target users during the search process. Additionally, the choice of the query approach is usually based on literature reviews and previous studies confirming its strengths. However, this assessment and these studies are often focused only on this specific approach with no comparison with the rest of the available approaches. For instance, NL-based approaches, such as PowerAqua and FREyA, were shown to be highly effective as well as satisfying [LMU06, DAC10] with no comparison with view-based approaches (especially with respect to their weaknesses such as the support provided for users during query formulation). Similarly, view-based approaches such as K-Search and Affective Graphs showed positive results and appreciation from the users who evaluated them [BCC+08, SDE+13], with no comparison with NL-based approaches (especially with respect to their weaknesses such as their ease of use and efficiency). This is an acknowledged problem due to these user-based evaluations being logistically complex, requiring careful organising and scheduling, and having high overheads including resource allocation, subject recruitment, evaluation organisation, system preparation and data analysation of the results. Despite this, the necessity of conducting these evaluations and comparisons before attempting to design a user-oriented approach is self-explainable. The rest of this section presents the requirements for designing the proposed query approach which are categorised into *functional* and *non-functional* requirements as follows:

6.1.1 Functional Requirements

The list of requirements discussed below are gathered from my understanding of the literature of semantic search; the current state of this research area and of the Semantic Web in general; as well as current challenges facing semantic search approaches.

- Approach should be open-domain/domain-independent: Earlier semantic search systems were highly domain-dependent, either being developed for a specific domain and application (requiring high customisation efforts to be portable across domains) or allowing access to a single domain at a time. However, to make full use of the potentials of the Semantic Web and existing datasets in answering a wider range of users queries, current efforts and the proposed approach should allow querying information spanning multiple heterogenous domains.

- Approach should allow accessing large semantic repositories of high complexity: Related to the above requirement, and to be kept up-to-date with change in the Semantic Web (rather than only being able to query datasets of small size or simple structure), the approach should allow accessing large and highly complex datasets especially with regards to their structure.

- Approach should allow bridging the gap between users and the system: An acknowledged problem facing text-based (semantic) search systems (adopting keywords or NL as query format) is the gap between them and their users caused by the latter's lack of knowledge of the exact data model. This gap results in users

using their own query terms, which are most often different from those found in the data and understandable by the system. The approach should attempt to bridge this gap either by supporting users in knowing the data structure or supporting the system in understanding the users' query terms.

- Approach should allow handling ambiguities: Usually, determining the concept denoted by a user query is not straightforward; ambiguities often arise. Different approaches provide various levels of support for users while tackling this challenge. For instance, while some may take complete responsibility of automatically resolving such ambiguities, others require help from the user and engage them in this process.

- Approach should allow hiding complexities from the user: Some approaches require users to learn a specific query language or certain expressions to be able to query the underlying data. However, this places a burden on users who wish to focus on finding satisfactory answers without recquiring additional learning. Thus, the approach should allow users to perform their search tasks in a natural manner, with little training required.

- Approach should allow both expert and casual users to retrieve data: Although several systems within the Semantic Web community are currently only used by experts in the field, the ultimate goal for the Semantic Web and semantic search is to reach a wider population of users. Therefore, the approach should be usable by both types of users, catering for their different needs and preferences with little or no training required.

6.1.2 Non Functional

- Approach should be effective: As stated above, a (semantic) search system assists users in fulfilling their information needs. Therefore, the query approach should achieve effectiveness by allowing users to successfully retrieve the required answers for their needs.

- Approach should be efficient: Efficiency of a search system is an important factor that influences users' overall success and satisfaction. Therefore, in addition to being effective, the approach should allow efficient querying of the underlying data. This minimises the resources expended in relation to the accuracy and completeness with which users achieve goals.

- Approach should be usable: There are several aspects which can affect the usability of a query approach. Firstly, the approach should be easy to use and intuitive, requiring minimal training and learning. Secondly, the approach should provide high support for users during query formulation. Different strategies are adopted to address this requirement, most addressing how to inform users of the possible queries which can be posed to the search system. Finally, the query language adopted should provide a high level of expressiveness. The latter specifies what

queries a user is able to pose and thus influences users' ability in formulating their information needs and their satisfaction with the query approach.

6.2 Requirements For User-Based Evaluations

As noted above, in order to develop a user-oriented semantic search query approach it is necessary to understand how current SOA query approaches are perceived by the target users, and whether they satisfy their needs. At the time of conducting this work, several semantic search evaluation initiatives were initiated, including the SemSearch challenge [HHM+10], the QALD open challenge [UCLM11] and the TREC Entity List Completion task [BSdV10]. Regardless their different goals, target systems and evaluation tasks, all of these initiatives were limited to assessing the retrieval performance of different semantic search systems, with a lack of user-related aspects (such as usability and satisfaction). The only work which evaluated and compared different query approaches in a user-based study was conducted by Kaufmann in her thesis "*Talking to the Semantic Web – Natural Language Query Interfaces for Casual End-Users*" [Kau07]. Kaufmann conducted a within-subjects evaluation of four semantic search prototypes adopting NL- and graph-based approaches with 48 casual users. The differences between this evaluation and the ones required for answering my research question are the following: 1) the evaluated query approaches did not include the form-based approach; 2) the evaluation investigated the perception of casual users only (and not expert users); 3) the evaluation did not use real-world queries, and the queries used did not cover some features (such as comparatives, negation, high degree of complexity); and finally, 4) the evaluation did not include assessing the extended learnability of the different approaches. Based on the above, it was decided to conduct two user-based evaluations as part of this thesis to answer the research questions previously outlined: "*How do casual and expert users perceive the usability of different semantic search query approaches?*", and "*Can training and frequency of use of a query approach improve the proficiency level and efficiency of users (in terms of time and effort) in answering search tasks of different complexity?*".

The rest of this section presents the requirements for designing these evaluations.

6.2.1 Requirements

As discussed in Section 4.4.2, the common procedure for user-based studies includes the following steps:

1. Assign participants various 'realistic' tasks to perform.
2. Take quantitative measurements of 'performance' (e.g. time taken, number of tasks completed, number of errors made, etc.).
3. Make observations about how the interface/system is being used by the participants.
4. Collect subjective reactions from the participants (e.g. satisfaction, usability).

Several design choices followed in performing these steps (such as dataset selection, evaluation criteria, subjects recruitment, etc.) can affect the reliability and results of an evaluation. Therefore, these choices should be based on a set of requirements carefully identified. The requirements presented below are based on a thorough review of evaluations in IR, discussed in Chapter 4.

6.2.1.1 Dataset

The main requirements for the choice of the dataset to be conformed to are listed below:

- To contain – and thus represent – real data found on the Semantic Web – in terms of data quality, heterogeneity and noise.
- To be of large-yet-manageable size, and to provide a balance between realism in the study and the ability of the evaluated systems and users to work with it.
- To be, or to contain a subset of, a simple and understandable domain for the recruited subjects to be able to reformulate the evaluation questions into the systems' query language without having problems understanding them.
- To be widely-known and frequently used within the community.

6.2.1.2 Queries

The main requirements for the choice of the queries, to be conformed to, are listed below:

- To be real-world queries describing genuine user information needs, as opposed to artificial queries created to simulate the first.
- To comprise different levels of complexity (in terms of number of concepts and properties and features) and different features (such as comparatives and superlatives).
- Whilst conforming to the above requirement, care should be taken to have queries which are not very difficult or complicated, and can be easily understood and reformulated by the recruited subjects (feeling natural and similar to queries they are used to).
- To be of a number sufficient enough to ensure representativeness, coverage and reliability of the evaluation, while balancing this with its effects on other aspects of the evaluation such as the required resources (in terms of time and cost) and tiredness of the recruited subjects.

6.2.1.3 Criteria, Measurements and Data Collection

As discussed in Section 4.4.1, several criteria and measures (other than retrieval effectiveness) have been used in literature to assess how well users achieve their goals and find satisfaction with a specific search system. The ones used in the evaluations described in this thesis are listed below:

- Effectiveness: Rather than assessing the performance of the search system in retrieving the required answers (for instance, through the use of precision and recall), effectiveness (of the query approach) is subjectively assessed by the subjects formulating the search tasks to show whether they could use the approach to successfully fulfill their needs.

- Efficiency: Efficiency has been one of the main evaluation criteria in user-based studies conducted in IIR and in HCI. Its most common measures are time and effort-based since they can inform the amount of resources required by the subjects for achieving their goals (according to the ISO definition).

- Satisfaction: Since the work in this thesis focuses on the users' perceptions and needs, a main criterion to assess is their satisfaction with the evaluated approaches and systems. Satisfaction is usually assessed subjectively using questionnaires or interviews with the subjects.

- Usability: This is the main criterion and the focus of the evaluations described in the thesis. Usability (as perceived by the subjects) is influenced by several factors including the ease of use of the query approach, its support during query formulation, the expressiveness of the query language adopted, as well as the criteria listed above (effectiveness, efficiency and satisfaction). Therefore, it is informed by the results of measuring all the above in addition to user questionnaires.

Finally, the above criteria should be measured using a combination of both objective and subjective data to provide a complete picture and allow for deeper analysis.

6.2.1.4 Experiment Setup

In conducting the user-based evaluations described here, care should be taken to conform to the following requirements related to the execution of the experiment.

- The evaluation should be carried out in a laboratory setting to have control over the environment and the experiment variables which may affect the outcomes of the evaluation. This is necessary since these evaluations attempt to answer specific questions and examine the effect of specific variables on others (such as the effect of the type of user or type of query approach on the results).

- One of the requirements of the query approach described in this work is to be usable by both expert as well as casual users, based on the ultimate goal of the Semantic Web, and to be accessible and reachable by both types of users. Therefore, the evaluations which are intended to assess current SOA query approaches should include both types of users to identify their different needs and preferences and to understand their perceptions. Similarly, to be representative of the target population and to increase the reliability and realism of the evaluation of the proposed query approach, both types of users should be recruited.

- To balance reliability of the evaluations with the amount of resources and feasibility of conducting them, between 8 and 12 subjects should be aimed at.

- Systems/prototypes used in the evaluations should be carefully selected or developed to have the required features and characteristics.
- Care should be taken to guarantee the availability of sufficient resources required to conduct the evaluations. Resources include: the cost, time and effort required to recruit subjects; the organising, scheduling and running of the experiments; and the analysis of the resulting data. These resources increase with the number of evaluated systems as the process gets more complicated.

In addition to the above requirements, specific to the usability study is the need to have systems/prototypes which allow evaluating the different query approaches (free and guided NL-based, graph-based, and form-based). Ideally, it would be beneficial to evaluate more than one system adopting a specific approach. The risk when evaluating only one system is the high influence it could have on the results, since its own strengths or weaknesses (due to specific implementation details as opposed to approach-related ones) could be falsely related back to the approach. However, it is important to note that it could be highly difficult to find such systems which are actively developed and managed, let alone to be able to have access to them to do any preparations required for the evaluation.

Furthermore, specific to the learnability study is the need to evaluate the *extended learnability* of the best performing query approach, as opposed to its *initial learnability*. The first focuses on assessing the change in performance over time, which is required here. This places a requirement that the evaluation is conducted over an extended period of time, usually through several sessions with fixed intervals of time in-between. It is thus important to decide on the number of sessions as well as how they are implemented, since this could directly influence the feasibility of recruiting subjects (which is always a difficult process, even for a one-session evaluation) as well as the amount of resources required for the evaluation.

6.3 Design Choices – Addressing the Requirements

Several design choices were adopted to conform to the above requirements while answering the thesis research questions. The rest of this section describes my solutions in addressing these requirements in each step.

6.3.1 Design Choices For A User-Oriented Semantic Search Query Approach

In order to adhere to the requirements listed above, I have developed a solution (named NL-Graphs) with the following design choices: firstly, NL-Graphs is domain-independent and, although being tested with DBpedia, can be configured to query different datasets spanning multiple domains. Since NL-Graphs adopts a hybrid query approach (as will be described in Chapter 9), it contains two main components: a graph-based and a NL-component. The graph-based component is configured to query either local or remote

SPARQL endpoints, while the NL-component only requires building an index for the ontology describing the dataset. With respect to large and complex datasets, using the NL-component allows both users as well as the graph-based component to reach and focus on a specific point in the dataset – thus, much smaller and simpler. Indeed, DBpedia, which is used in evaluating the approach, is an example of such large, heterogeneous and complex datasets.

To bridge the gap between the user and the system, NL-Graphs attempts to accomplish this through two complementary solutions. Firstly, query expansion (described in Chapter 9) is performed when no matches are found in the underlying ontology for a query term or when no results are generated using the identified matches. Note that the terms used in the query expansion process are gathered from BabelNet, a recently published knowledge base which benefits from wide-coverage resulting from its integration of WordNet and Wikipedia. If query expansion also fails to return interpretations for one or more query terms, the user has the ability to directly select the intended ontological terms through the visualisation of the dataset, provided by the graph-based component.

With respect to ambiguities, which are normal to occur in a user's query, several techniques are designed within NL-Graphs in order to help resolve them. The first is to perform *automatic disambiguation* using a word sense disambiguation (WSD) approach, developed specifically for this task and described in Chapter 9. Then, depending on the output of the WSD, the interpretations of the NL-component for all query terms are shown to the user. The second is through *user-engagement*, since users can choose to accept some or all of the generated matches. However, it is possible in certain scenarios that the disambiguation and, as a result, the interpretation of a specific query term, is not satisfactory for the user. Here, the solution would be to use the third technique, which is *query refinement*. To refine the query, the user can either change the NL query or choose to visually construct it using the graph-based component. Indeed, showing the interpretations of the system acts as feedback for the users and help them understand which parts of the query *failed* and require refinement. Otherwise, they would need to continuously perform random trials/reformulations.

Moreover, related to the requirements with respect to ease of the query language and allowing both casual and expert users to use NL-Graphs, the NL-component provides the means for a straightforward method for query formulation where users are free to enter their queries in the natural language that they are most used to. Thus, users are not required to learn a specific query language. Additionally, they do not need to acquire specific domain knowledge or expertise since they can understand the underlying data and the possible ways to query it through the visualisation provided by the graph-based component.

Finally, regarding the non-functional requirements, the first is for NL-Graphs to be effective. By doing all the above, the approach is attempted to allow users to successfully retrieve the required answers for their needs and thus achieve the required effectiveness. Secondly, an attempt to balance support during query formulation and increased query language expressiveness with the effort and time required, the graph-based approach

is combined with the NL-based one. The latter is intended to increase the efficiency of NL-Graphs by providing the means for an easy-and-fast starting point for query construction.

As stated earlier, the usability of the query approach is affected by the expressiveness of the query language adopted, the support provided for users during query construction and the ease of use of the query approach. The expressiveness of the query language is achieved through several design choices: firstly, by allowing users to enter free-form NL queries consisting of keywords, phrases or full questions. Secondly, the graph-based component provides another alternative to construct the query by visual means, which is intended to increase the users' ability to pose a wider range of queries with various levels of complexity to fulfill their information needs. Additionally, the visual approach increases the support for users during query construction since it provides an understanding of the available data and how it is structured and possible ways of querying it. Moreover, the feedback provided for the users which shows the system's interpretation for their queries is another means of support. For instance, this feedback informs users which parts of the query succeeded and which parts failed or require refinement. Finally, the ease of the use of the query approach is achieved through all of the above, in addition to adopting a NL input feature which is intended to provide an easy and direct mechanism for constructing a query.

6.3.2 Design Choices For User-Based Evaluations

6.3.2.1 Dataset and Queries

In the usability evaluation, five semantic search prototypes were selected to evaluate the different query approaches. This placed a constraint on the choice of the dataset since it required verifying the prototypes' ability to work with the selected dataset. The only dataset which was available to use at this time and was manageable by all the prototypes was the geography dataset within the Mooney Natural Language Learning Data. It conforms to the requirement of being from a simple and understandable domain to avoid difficulties in understanding the data or formulating the evaluation questions by the recruited subjects. Additionally, it is well-known and frequently used within the community. However, it does not conform to the size and representativeness requirements since it is very small compared to other datasets found on the Semantic Web, in addition to being specially-created (usually of higher quality than operationally-derived ones). Moreover, with respect to the queries used, the dataset already contained more than 800 NL questions for which the groundtruth was also available. These questions were collected from university students as well as real users (volunteers) through a Web interface. Indeed, this does not conform to the first criterion (being real-world queries). However, it is a better alternative for simulating real information needs, since questions are given by real users as opposed to a set of evaluation organisers or experts. Additionally, as will be discussed in more detail in Chapter 7, this choice conforms to all the other criteria: for the queries to be easily understandable by the recruited subjects whilst covering different levels of complexity and allowing a sufficient number without

overwhelming the subjects.

In the learnability evaluation, only the best-performing system (Affective Graphs) is included. Therefore, conforming to the requirements of the dataset should be easier. Indeed, several datasets such as *Geonames, MusicBrainz, DBpedia* and *Sindice 2011* conformed to the size and representativeness requirements and were manageable by Affective Graphs. However, according to the requirements listed above with respect to the choice of queries, they ought to describe real-world information needs. At the time of conducting this evaluation, only query logs for DBpedia and SWDF datasets were made available by the USEWOD2011 data challenge[1]. On one hand, using a generic multi-domain dataset such as DBpedia would allow conducting the evaluation with both expert and casual users (which is another requirement, as shown above), since it would be possible to use queries from an easy and understandable domain. However, experimentations showed that, due to the complexity and structure of DBpedia, formulating queries using a graph-based approach would be very difficult (if not impossible) for users who do not have sufficient knowledge of the dataset and its structure. Therefore, the SWDF dataset was chosen for this evaluation, which also influenced the type of users recruited as described below. Referring back to the queries used in this evaluation, they are real-world queries, comprising different levels of complexity and features (such as comparatives and superlatives), and all attempts were made to ensure that they can be understood and reformulated by the recruited subjects.

Similarly, in the evaluation of the hybrid query approach (developed based on the outcomes of the above evaluations), the choice of the queries influenced the choice of the dataset. To allow assessing the usefulness of the hybrid approach, it was necessary to find a set of queries with which NL-based approaches would face problems while attempting to answer. These problems would be resulting from the difficulty of mapping user query terms to ontological ones or understanding complex questions such as those containing comparatives, superlatives or advanced constraints. Fortunately, these queries could be selected from the data provided by the Question Answering over Linked Data (QALD) workshop in its challenges. In each challenge, a set of questions written by university students were made available for both DBpedia and MusicBrainz datasets. DBpedia was therefore selected since it is more suitable in conforming to the requirements listed above. Similar to the usability evaluation, these questions might not exactly describe real-world scenarios (not collected from operational search engines, for instance). However, it could be argued that university students could be seen as potential/real users for this data. And again, the queries selected for the evaluation cover different levels of complexity while being understandable by both types of users.

6.3.2.2 Criteria, Measurements and Data Collection

In the three user-based evaluations described in this thesis, the same criteria and measures have been used, since they all focus on usability-related aspects. Firstly, to measure effectiveness of a specific query approach and how well it allowed users to achieve their goals and answer their information needs, success rate – capturing the percentage of

[1] http://data.semanticweb.org/usewod/2011/challenge.html

tasks successfully completed – is used. Secondly, input time required by users to formulate their queries in addition to the number of attempts (query reformulations) required by users for a specific query are used as effort-based measures for assessing efficiency. Logs are used – as a data collection method – to collect this objective data which is automatically gathered using custom-written software to allow each experiment to be orchestrated.

Questionnaires, as well as observations, are used to collect subjective data. To measure usability and satisfaction, two post-search questionnaires are included in each evaluation. The first one is the *System Usability Scale (SUS) questionnaire* (see Figures A.3 and A.4 in Appendix A). It is a standardised usability test consisting of 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. The subjects answer all questions on a 5-point Likert scale identifying their view and opinion of the system. A SUS score of 0 implies that the user regards the interface as unusable, and a score of 100 implies that the user considers it to be perfect. A score of approximately 60 and above is generally considered as an indicator of good usability. The second questionnaire (*Extended Questionnaire*: see Figures A.5 and A.6 in Appendix A) is one which was designed to capture further aspects of the users' satisfaction, specific to each evaluation, and thus contained different questions in each of them. For instance, in the usability evaluation, overall questions (presented after evaluating all the approaches) were included asking the user to rank the systems with respect to certain aspects such as *how much they liked the query approach adopted*. These questions were intended to allow more accurate comparisons, since rankings are an inherently relative measure, while individual questionnaires are completed after evaluating each system's evaluation (and thus temporally spaced) with no direct frame of reference to any of the other systems. Additionally, this questionnaire contained open-ended questions to gather additional feedback from the users regarding their experience, satisfaction and preferences. Open-ended questions are acknowledged to help researchers understand the users' rationale for answering closed-ended questions in a specific way. Moreover, a third questionnaire was used to collect demographic data such as age, profession and knowledge of linguistics (see Figures A.7 and A.8 in Appendix A). More details about the questionnaires are provided in Chapters 7, 8 and 9 and in the appendices. Finally, experiment leaders were always present for observations and feedback purposes, which is acknowledged to recognise issues that would otherwise be ignored.

6.3.2.3 Experiment Setup

To conform to the previously-listed requirements, the three evaluations took place in a controlled laboratory setting with one or more experts present for the whole length of the evaluation. Additionally, in both the usability study and the evaluation of the hybrid query approach, casual as well as expert users were recruited. Casual users refer to "those with very little or no knowledge of the Semantic Web", and expert users to "those who have knowledge and experience in the Semantic Web". However, as discussed above, the dataset used in the learnability evaluation (SWDF) influenced the

choice of the users. The SWDF dataset contains information on publications, people, and organisations that were part of the main conferences and workshops in the area of Semantic Web. Therefore, it was decided to use only expert users for this evaluation since casual users would not be familiar with the domain and could face difficulties understanding and reformulating the selected queries. With respect to the number of subjects, 10-12 subjects (of one specific type: 10 casual users and 10 expert users) were always recruited for each evaluation. Moreover, the usability evaluation followed a within-subjects design (in which all recruited subjects evaluate all the approaches) to allow direct comparison between the evaluated query approaches to increase the reliability and usefulness of the results. Details of the recruitment process is provided in the relevant chapters (See Chapters 7, 8 and 9).

As discussed above, it is difficult to find systems within the Semantic Web community which are actively developed and managed. Most of the time these are prototypes built for a specific project or as part of a PhD student's research work, and stop being developed when these come to an end. Therefore, a long process of analysing systems built for semantic search and trying to reach their developers resulted in having access and permission to work with only five of them (to be included in the usability evaluation). These (and the adopted query approach) are: NLP-Reduce (free-NL), Ginseng (controlled-NL), Semantic-Crystal (graph-based), K-Search (form-based), and Affective Graphs (graph-based). The first three systems were developed as part of a thesis at the University of Zurich [Kau07], while the last two were developed as part of two different PhD students' work at the University of Sheffield [BCC+08, SDE+13]. To prepare them for the evaluation, the systems' ability to access and query the evaluation dataset (Mooney) was verified. Then, a wrapper was specifically developed for each system to connect it with the evaluation software which was built to orchestrate the whole experiment. This included showing the instructions for the users, gathering various forms of data (timings, queries issued, etc.) and requesting input for the evaluations' questionnaires from the users.

Finally, with respect to the execution of the evaluations, the process proved to be time-consuming and cost-intensive. For instance, in the usability evaluation, each full experiment with one user took between 60 to 90 minutes, with 20 subjects involved: approximately 30 hours of subject time alone. Since each subject was chaperoned at all times for feedback purposes, an equal amount of time was spent running the evaluation. Note that, care was taken so as not to affect users' behavior or experience by having test leader(s) observing them. As explained in the instructions sheet provided for subjects before starting each evaluation (see Appendix A, B and C), subjects were recommended to consult the leader in case of questions and problems related to problems with completing the experiment, as opposed to help on answering the evaluation questions.

This excludes additional logistical time and effort for organising and scheduling the evaluation. Indeed, this effort was much more for the learnability evaluation (which required three sessions over three days), since the organisation process was more complicated compared to other evaluations (which take place in one session). Furthermore,

there was the overhead incurred in the data analysis, since in user-based evaluations subjective data is as important as the objective data collected, and, as acknowledged, analysing this data is extremely time-consuming and labor-intensive [Kel09].

6.4 Summary

This chapter has described the requirements and design choices of a user-oriented semantic search query approach. In order to achieve this, it is important to understand the target users' needs, preferences and their perceptions of current query approaches. Therefore, a user-based evaluation is first conducted to investigate the latter and understand how both expert and casual users perceive the usability of SOA query approaches. The best performing approach (and the system adopting the approach) is then selected for a user-based evaluation to investigate its learnability and the effect of training and frequency of use on users' efficiency. The requirements of the query approach and of these two evaluations have been presented. Then, the design choices followed in developing the query approach and running the evaluations, in order to conform to the requirements, have been discussed. The following chapters describe the design of these evaluations in detail and their most interesting outcomes. Based on these outcomes, a proposed query approach and its implementation are described in a later chapter.

Chapter 7

Evaluating Usability of Semantic Search Query Approaches

As discussed in Chapter 3, semantic search approaches employ different query formats. These formats offer different levels of expressiveness and support for users during query formulation. This chapter presents the user-based study conducted to understand how (expert and casual) users perceive the usability of different semantic search query approaches. The main purpose of this study is to understand users' needs and requirements for designing a more suitable and user-oriented query approach that would provide support for users during query formulation, together with a satisfying level of usability and effectiveness. The study also provides a first-time understanding and comparison of how the two types of users perceive the usability of these approaches. Developers of future query approaches and similar user interfaces, who have to cater for different types of users with different preferences and needs, could highly benefit from the results and findings of this study.

The rest of the chapter is structured as follows: Section 7.1 provides an overview of the usability study and describes the methodology adopted, with information about the dataset used and the setup of the experiment. Section 7.2 describes the analysis performed on the collected data, then discusses the results of comparing the different query approaches. Finally, Section 7.3 presents the main conclusions of this work.

7.1 Evaluation Design

The underlying question of this study is how users perceive the usability of different semantic search query approaches, and whether this perception is different between expert and casual users. To answer the question, ten casual users and ten expert users were asked to perform five search tasks with five tools adopting NL-based and view-based query approaches. As discussed in Section 6.3.2, the selected tools for this evaluation

(and their adopted query approaches) are: NLP-Reduce (free-NL), Ginseng (controlled-NL), K-Search (form-based), and Semantic-Crystal and Affective Graphs (graph-based).

As discussed in Section 6.2, both objective as well as subjective data are collected during the experiment to assess effectiveness, efficiency, satisfaction and usability. The time required by users to formulate queries using a specific tool's interface and the number of attempts required for each query are used to measure efficiency, while users' success rates in finding satisfying answers for the given queries informs effectiveness of the approach. Finally, questionnaires were used to measure usability and satisfaction. They included questions that assessed users' satisfaction with each tool in separate, as well as questions which required users to rank the tools according to specific criteria such as: *how much they liked the tools* or *how much they liked their query interfaces.*

7.1.1 Dataset and Questions

According to the design choices presented in Section 6.3.2, the geography dataset within the Mooney Natural Language Learning Data was selected for this evaluation. The original Prolog knowledge bases were translated into OWL by [Kau07]. The resulting geography OWL knowledge base contains 9 classes, 11 datatype properties, 17 object properties and 697 instances.

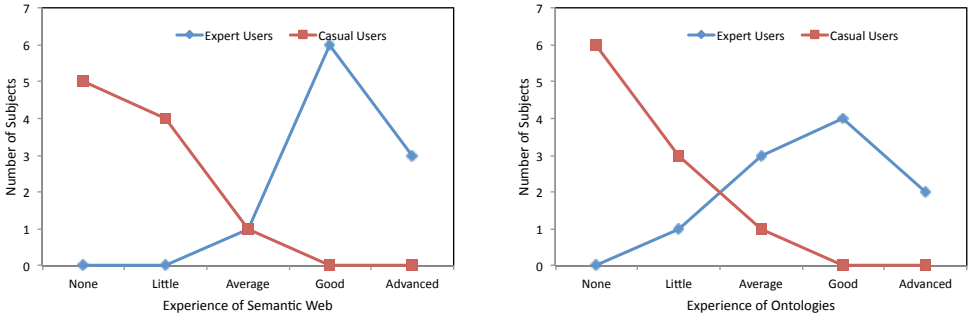
The dataset contained a set of English questions, which were composed by undergraduate students of the computer science department of the University of Texas and gathered from 'real' people using a Web interface provided by Mooney's research group. There exist 877 natural language questions for the geography knowledge base. For each question, there can also be found a corresponding logical representation in the dataset formatted as Prolog terms. These questions were used as templates to generate the queries used in this study, for which the groundtruth was also available with the question-set. Five evaluation questions ranging from simple to complex were chosen to test each tool's ability in supporting specific features such as comparison or negation (to inform expressiveness).

1. *Give me all the capitals of the USA?*
   This is the simplest question: consisting of only one ontology concept: '*capital*' and one relation between this concept and the given instance: *USA*.
2. *What are the cities in states through which the Mississippi runs?*
   This question contains two concepts: '*city*' and '*state*' and two relations: one between the two concepts and one linking *state* with *Mississippi*.
3. *Which states have a city named Columbia with a city population over 50,000?*
   This question features comparison for a datatype property *city population* and a specific value (50,000).
4. *Which lakes are in the state with the highest point?*
   This question tests the ability for supporting superlatives (*highest point*).
5. *Tell me which rivers do not traverse the state with the capital Nashville?*
   Negation is a traditionally challenging feature for semantic search [DAC10, LFMS12].

7.1.2 Evaluation Setup

As illustrated in Section 6.3.2 and to conform to the evaluations' requirements, it was decided to recruit 20 subjects (10 expert users and 10 casual users) in order to strike a balance between the reliability of the evaluation and its overhead. The casual users were drawn from the staff and student population of the University of Sheffield after the usability study was promoted on its relevant mailing lists. On the other hand, the expert users were drawn from the Organisations, Information and Knowledge (OAK) Group[1] within the Department of Computer Science at the University.

Figure 7.1 shows a clear distinction between the two types of users in their knowledge of the Semantic Web and ontologies.



(a) Semantic Web

(b) Ontologies

Figure 7.1: User experience of the Semantic Web and ontologies.

The 20 subjects (12 females, 8 males), aged between 19–46 with a mean of 30 years, were rewarded for their time. The experiment took place in a controlled laboratory setting following a within-subjects design to allow direct comparison between the evaluated query approaches. Typically, with this design less participants are required to get statistically significant results [ATT10]. All 20 subjects evaluated the five tools in randomised order to avoid any carryover/order effects (discussed in Section 4.4.2.2), that could influence the experiment results. Furthermore, to avoid any possible bias introduced by developers evaluating their own tools, only one test leader – who is also not the developer of any of the tools – was responsible for running the whole experiment. This guaranteed fairness of the evaluation process; tools were explained in equal time periods, in the same way and by the same person for each subject.

For each tool, subjects were given a short demo session briefly explaining the query language adopted by the tool and – through an example – how to use it to formulate a sample query. After that, subjects then proceeded to the actual experiment in which they were asked to formulate each of the five questions in turn using the tool's interface. The order of the questions was randomised for each tool to avoid any learning effects. After testing each tool, subjects were asked to fill in two questionnaires to capture their experience and level of satisfaction. Finally, after evaluating the five tools, they were presented with a questionnaire to collect demographics data such as age, profession and

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

knowledge of linguistics, among others (see Figures A.7 and A.8 in Appendix A). Each full experiment with one user took between 60 to 90 minutes.

In accordance with the design choices presented in Section 6.3.2, and in order to have a complete picture and allow for deeper analysis in order to measure the required criteria, both objective and subjective data were collected covering the experiment results. To measure efficiency, the *input time* required by users to formulate their queries in the respective tool's interface as well as the *number of attempts* showing how many times on average users reformulated their query were recorded. Additionally, the *success rate*, capturing the percentage of tasks successfully completed, was used to measure effectiveness. Finally, subjective data collected through two post-search questionnaires was used to measure usability of the approaches and satisfaction of users. The first is the *System Usability Scale (SUS) questionnaire* (see Figures A.3 and A.4 in Appendix A), used to investigate usability, while the second is the *Extended Questionnaire* (see Figures A.5 and A.6 in Appendix A), which captured further aspects of the users' satisfaction with respect to the query approaches adopted and the content returned in the results, as well as how it was presented. After completing the experiment, subjects were asked to rank the tools according to four different criteria (each one separately). These criteria are: how much they liked the tools (*Tool Rank*); how much they liked their query interfaces: graph-based, form-based, free-NL and controlled-NL (*Query Interface Rank*); how much they found the results to be informative and sufficient (*Results Content Rank*); and finally how much they liked the results presentation (*Results Presentation Rank*). Note that users were allowed to give equal rankings for multiple tools if they had no preference for one over the other. To facilitate comparison, for each criterion, ranking given by all users for one tool was summed and subsequent score was then normalised to have ranges between 0 and 1 (where 1 is the highest).

7.2 Results and Discussion

Results for both expert and casual users are presented in Tables 7.1 and 7.3 respectively. In these tables, a number of different factors are reported such as the SUS scores and the tools' rankings (explained above). *EQ1: liked presentation* shows the average response to the question "I liked the presentation of the answers" given in the extended questionnaire and scored out of the 5-point Likert scale. *EQ2: information sufficient* shows the average response to the question "The information given in the answers was sufficient", and *EQ3: query language easy* shows it for the question "The system's query language was easy to use and understand. The *Number of attempts* shows how many times on average the users reformulated their query using the tool's interface in order to obtain answers with which they were satisfied (or indicated that they were confident a suitable answer could not be found). This latter distinction between finding the appropriate answer after a number of attempts and the user 'giving up' after a number of attempts is shown by the *Success rate*, which is averaged over the 5 questions. *Input time* refers to the amount of time the subjects spent formulating a query using the tool's interface before submitting it.

Table 7.1: Tools results for expert users. Non-ranked scores are median values; bold values indicate best performing tool in that category.

Criterion	**AG**	**SC**	**K-S**	**Gins.**	**NLP-R**	**p-value**
SUS (0-100)	**63.75**	50	40	32.5	37.5	0.003
Tool Rank (0-1)	**0.875**	0.625	0.6	0.225	0.225	-
Query Language Rank (0-1)	**0.925**	0.725	0.65	0.425	0.45	-
Results Content Rank (0-1)	0.875	0.875	**0.925**	0.725	0.725	-
Results Presentation Rank (0-1)	0.875	0.875	**0.975**	0.8	0.8	-
EQ1: liked presentation (0-5)	2.5	2.5	**4**	3	3	0.007
EQ2: information sufficient (0-5)	**4**	3	2.5	1.5	1.5	0.001
EQ3: query language easy (0-5)	**4**	**4**	**4**	2	2.5	0.035
Number of Attempts	**1.5**	2.2	2	1.7	4.1	0.001
Success Rate (0-1)	**0.8**	0.4	0.5	0.4	0.2	0.004
Input Time (s)	88.86	79.55	53.54	102.52	**19.90**	0.001

Note that in the rest of this section, the term *tool* (e.g. graph-based tools) is used to refer to the implemented tool as a full semantic search system (with respect to its query interface and approach, functionalities, results presentation, etc.), while the term *query approach* (e.g. graph-based query approach) is used to specifically refer to the style of query input adopted.

The data collected during the evaluation was quantitatively and qualitatively analysed. To quantitatively analyse the results, SPSS[2] was used to produce averages, perform correlation analysis and check the statistical significance. The median (as opposed to the mean) was used throughout the analysis as the main measure of central tendency, since it was found to be less susceptible to outliers or extreme values sometimes found in the data. In the qualitative analysis, the open coding technique [SC90] was used, in which the data was categorised and labelled according to several aspects dominated by usability of the tools' query approaches and returned answers.

7.2.1 Results for Expert Users

As explained in Chapter 6, five semantic search prototypes were selected to evaluate the different query approaches. These are Affective Graphs, Semantic Crystal, K-Search, Ginseng and NLP-Reduce. This selection was based on finding prototypes which are actively developed and managed or which I could have access to their implementations to do the required preparation for the evaluation. For instance, this led to excluding *Querix*, the fourth prototype developed and evaluated by Kaufmann in her thesis (besides Semantic Crystal, Ginseng, and NLP-Reduce).

Figure 7.2 shows the scores of the tools for the SUS questionnaire. In order to have a good understanding of the results, the box plot is used to show the median, quartiles

[2] www.ibm.com/software/uk/analytics/spss/

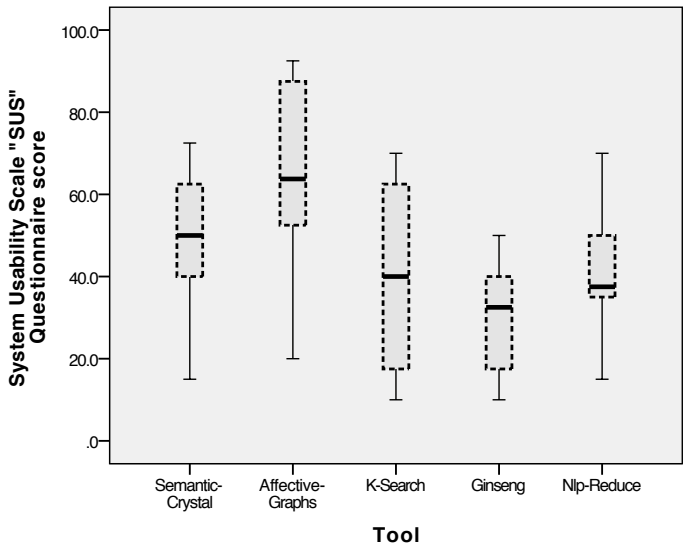
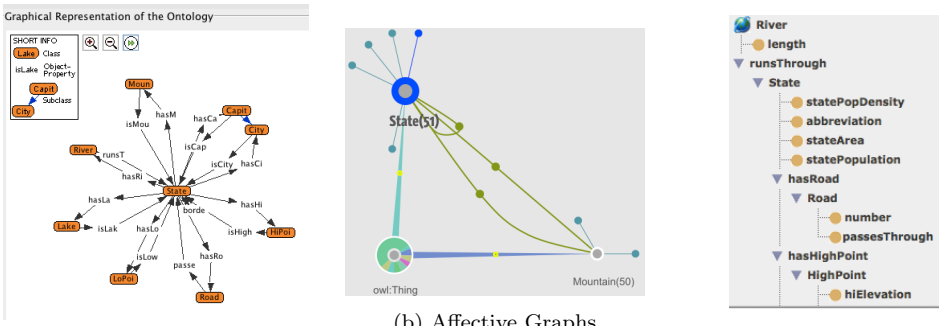


Figure 7.2: System Usability Scale (SUS) questionnaire scores for expert users

and range of scores received by the tools. The following abbreviations for tools' names are used for readability: AG for Affective Graphs, SC for Semantic Crystal, K-S for K-Search, Gins. for Ginseng and NLP-R for NLP-Reduce. Note that, although expert users where chosen from the same research group of the developers of Affective Graphs and K-Search, care was taken so as not to include any researcher who participated in designing, implementing or collaborating by any other means on these systems, to avoid any bias or influence on the results.

According to the adjective ratings introduced by [BKM09], Ginseng – with the lowest SUS score – is classified as *Poor*, NLP-Reduce as *Poor* to *OK*, K-Search and Semantic Crystal are both classified as *OK*, while Affective Graphs, which managed to get the highest average SUS score, is classified as *Good*. These results are also confirmed by the tools' ranks (see Table 7.1): Affective Graphs was selected 60% of the times as the most-liked tool and thus got the highest rank (0.875), followed by Semantic Crystal and K-Search (0.625 and 0.6 respectively) and finally Ginseng and NLP-Reduce got a very low rank (0.225) with each being chosen as the least-liked tools four times and twice, respectively. Since the rankings are an inherently relative measure, they allow for direct tool-to-tool comparisons to be made. Such comparisons using the SUS questionnaire may be less reliable since the questionnaire is completed after each tool's experiment (and thus temporally spaced) with no direct frame of reference to any of the other tools. Table 7.1 also shows that Affective Graphs, which was liked and found to be the most intuitive by users, managed to get satisfactory answers for 80% of the queries, followed by K-Search (50%), which employs the second most-liked query approach.

Note that, as explained above, in this study subjects were given hands-on training on the use of the query language adopted by each tool to formulate the evaluation queries. In fact, it could be interesting to investigate the difference in the results if this training was not provided. However, in my view, the absolute results provided by users (such as the SUS scores) could differ but not the relative rankings of the approaches and tools,



(a) Semantic Crystal (b) Affective Graphs (c) K-Search

Figure 7.3: Different visualisations of the Mooney ontology by the tools

especially that the same training is provided for all the subjects for all tools.

Expert users prefer graph- and form- based approaches

Results showed that graph- and form- based approaches were preferred by expert users. However, in terms of overall satisfaction (see Tool Rank in Table 7.1 and Figure 7.2), graph-based tools outperformed the form- and NL- based ones. This was an unexpected outcome of this study. The usability study by [KB07] with casual users found that the best-liked querying approach was the one allowing full English questions and the one least-liked was the graph-based approach since it was perceived to be too complicated and tedious to use. However, no similar studies evaluated how expert users perceive the usability of different querying approaches. This study is thus the first to show this. Although the graph-based tools took users more time to formulate their queries than with the form-based or the free NL one, this did not influence the users satisfaction with the tools or with the query interface by itself as shown in the scores given to the question *The system's query language was easy to understand and use* (see Table 7.1: EQ3). Additionally, feedback showed that *users were able to formulate more complex queries with the view-based approaches (graphs and forms) than with the NL ones (free and controlled)*. Indeed, the ability to visualise the search space provides an understanding of the available data (concepts) as well as connections found between them (relations) which shows how they can be used together in a query [KB07, ULL+07]. Figure 7.3 shows the visualisation of the Mooney ontology by the three view-based tools: Semantic Crystal, Affective Graphs and K-Search.

Although Affective Graphs and Semantic Crystal both employ graph-based query approaches, users had different perceptions of their usability. On one hand, Affective Graphs was the most liked tool (see SUS score and Tool Rank in table 7.1) and received the highest number of positive comments in the users' feedback. The most repeated (60%) positive comment was "*the query interface is intuitive and pleasant to use*". Again, this was a surprising outcome since graph-based approaches tend to be more complicated and laborious [KB07, ULL+07] than other approaches. On the other hand, the visualisation approach adopted by Semantic Crystal was preferred by users. As shown in Figure 7.3, Semantic Crystal visualises the entire ontology whereas Affective

Graphs opts for showing concepts and relations only selected by the users. Although users liked the first approach, it imposes a limitation on how much can be displayed in the visualisation window. The Mooney ontology used in this evaluation contains only nine concepts, which makes it very small compared to larges ontologies such as DBpedia. With this small ontology, the graph is clear and can be easily explored; as the ontology gets bigger, the view would easily get cluttered with concepts and links showing relations between them. This would negatively affect the usability of the interface and, in turn, the user experience. It is a big challenge to visualise and explore a large ontology in a graph-based approach; even though Affective Graphs only shows information (relations and connected concepts) about the concepts that the user selects, feedback showed that at least two users found that the view can get cluttered and affect their ability to formulate the required queries.

Expert users frustrated by controlled-NL

As shown in Table 7.1, Ginseng, which is employing a controlled-NL approach, was chosen as the least liked interface. As discussed in Section 3.3.3, Ginseng offers suggestions to the user according to a specific vocabulary and refuses entries that are not in the possible list of choices. Although this guidance through suggestions was at times appreciated – especially when the search space is unknown – restricting expert users to the tool's vocabulary was inhibiting. Expert users perceived this restriction as forcing them into following specific paths, which caused the query construction process to be very complicated and frustrating. Not allowing users to express their informations needs in their own words was also perceived by them as a drawback of Ginseng. These resulted in an unsatisfying experience (lowest SUS score of 32.5), which is supported by the most repeated *negative* comments given for Ginseng:

- *It is frustrating when you cannot construct queries in the way you want.*
- *You need to know in advance the vocabulary to be able to use the system.*

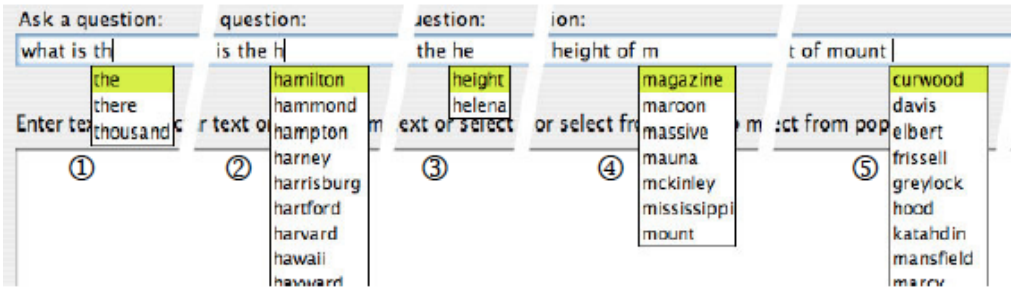


Figure 7.4: Ginseng query completion window [BKK05].

The second comment is in stark contrast to what the controlled-NL approach is designed to provide. It is intended to help users formulate their queries without having to know the underlying vocabulary, while at the same time preventing errors and mistakes. However, even with guidance, users frequently got stuck because they did not know how

to associate the suggested concepts, relations or instances together. This is illustrated in Figure 7.4, which shows an example of suggestions showed by Ginseng. As noticed, the user has to find a suitable choice from the suggestions for each of his own query terms. This outcome is confirmed by users requiring the longest input time when using Ginseng (Table 7.1 : Input Time).

Discussion of the results of individual SUS questions

Although the average SUS score given to a particular approach or tool is often used as a measure of its usability and learnability, some specific questions found in the questionnaire are more focused on these aspects and therefore could be interesting to present their results separately. Table 7.2 shows the results given by expert users for each tool for the following three questions:

- I thought the system was easy to use.
- I found the system very tedious / troublesome to use.
- I would imagine that most people would learn to use this system very quickly.

Table 7.2: Scores given by expert users for individual SUS questions. These questions are answered on a 5-point Likert scale ranging from *Strongly Disagree(1)* to *Strongly Agree(5)*. Bold values indicate best performing tool in that category.

Question (Strongly Disagree - Strongly Agree)	**AG**	**SC**	**K-S**	**Gins.**	**NLP-R**
System easy to use	**3.6**	2.6	2.3	1.5	2.5
System tedious to use	**2.1**	3	3.2	4.2	3.5
Learn to use the system quickly	**3.4**	2.6	2.2	1.8	3

Looking at the scores presented in Tables 7.2 and 7.1, the first observation is with respect to the results obtained by AG which managed to achieve the highest scores for the individual SUS questions, in line with the overall SUS scores and tools' ranks. It is also interesting to note the large difference between AG's scores and the rest of the tools ('3.6' versus '2.6', '2.1' versus '3' and '3.4' versus '3'), supporting the above findings which showed AG as the interface most liked most intuitive by users – easy to use and learn.

Similarly, in-line with the overall SUS scores and tools' ranks, Ginseng managed to obtain the lowest scores in the individual questions as shown in Table 7.2. This is fairly expected since the main reason for expert users not preferring Ginseng was related to its usability. Recall, the restriction to a specific vocabulary led to users struggling and having frustrating experience formulating their queries.

Another piece of result to observe here is with respect to NLP-Reduce which although was always ordered after AG, K-S and SC in the tools' rankings and overall SUS scores, obtained alternating scores with them in the individual questions especially the last one focusing on learnability. The explanation for this is that the free-NL approach adopted

by NLP-Reduce was appreciated by expert users for being simple, natural and thus not requiring much learning to use. Indeed, achieving low SUS score and being chosen as the least-liked tool is mainly due to the habitability problem and mismatch between users' terms and those understood by the system, leading to low success rates and high number of query reformulations (as will be discussed in Section 7.2.3).

7.2.2 Results for Casual Users

Table 7.3: Tools results for casual users. Non-ranked scores are median values; bold values indicate best performing tool in that category.

Criterion	**Affective Graphs**	**Semantic Crystal**	**K-Search**	**Ginseng**	**NLP-Reduce**	**p-value**
SUS (0-100)	55	**61.25**	41.25	53.75	43.75	0.485
Tool Rank (0-1)	**0.675**	**0.675**	0.575	0.45	0.275	-
Query Language Rank (0-1)	0.525	0.55	**0.625**	0.525	0.4	-
Results Content Rank (0-1)	0.675	0.75	**0.775**	0.575	0.575	-
Results Presentation Rank (0-1)	0.775	0.7	**0.8**	0.6	0.475	-
EQ1: liked presentation (0-5)	3	3	**3.5**	2.5	2	0.3
EQ2: information sufficient (0-5)	**3.5**	3.5	3	4	3	0.001
EQ3: query language easy (0-5)	**4**	**4**	**4**	3	3	0.131
Number of Attempts	**1.7**	1.8	2.1	**1.7**	4.2	0.001
Success Rate (0-1)	0.4	**0.6**	0.5	0.4	0.2	0.150
Input Time (s)	72.8	75.76	63.59	93.13	**18.6**	0.001

According to the adjective ratings given in [BKM09], K-Search and Nlp-Reduce -with the lowest SUS scores- are classified as Poor to OK, Ginseng, and Semantic Crystal as OK and Affective Graphs as Good. Some of these results are different than those given for the tools' ranks (see Table 7.3: Tool Rank). While Ginseng managed to get a higher SUS score than K-Search, users ranked it lower when they were asked to rank the tools according to how much they liked them. This is also illustrated in Figure 7.5 in which the upper quartile of the SUS scores given to Ginseng is lower than that of K-Search.

Although this is a conflicting result, it is interesting to note how tools perform when evaluated and assessed separately (reflected by the SUS score) and in comparison with other tools (reflected by the ranks). As explained above, these rankings are an inherently relative measure and therefore allow for direct tool-to-tool comparisons to be made. Such comparisons using the SUS questionnaire may be less reliable since the SUS questionnaire is completed after each tool's experiment (and thus temporally spaced) with no direct frame of reference to any of the other tools.

Looking at the results obtained by Kaufmann in her usability study with casual users, Ginseng – adopting a controlled-NL based approach– obtained a very similar SUS score (Kaufmann: '55.10' compared to '53.75') and received similar feedback as it was found to be assisting but also restrictive in its vocabulary. On the other hand, NLP-Reduce obtained a much higher SUS score (Kaufmann '56.72' compared to '43.75'), yet again the feedback is in line with our results: NLP-Reduce was perceived as the one with *the simplest interface* and *similar to common search engines* but its query language negatively perceived as *too relaxed* and not clear. Finally, the most difference is found in the results related to Semantic Crystal: – adopting a graph-based approach – a SUS

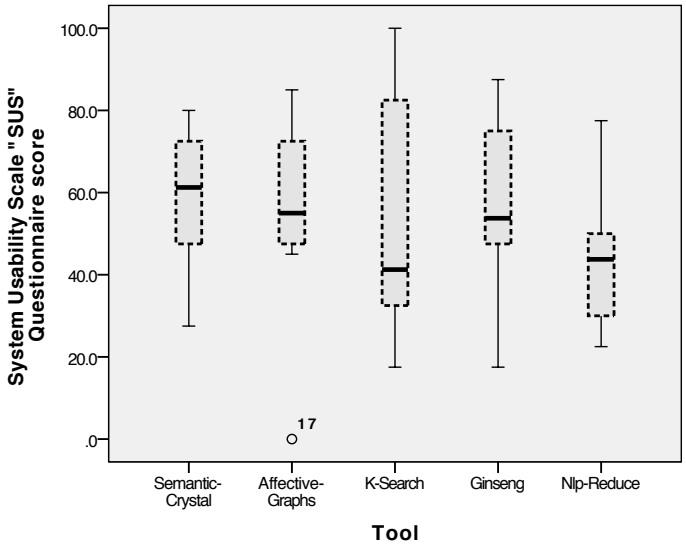


Figure 7.5: System Usability Scale (SUS) questionnaire scores for casual users

score of '36.09' given by Kaufmann's users compared to '61.25' given by our users, in addition to the interface being ranked as the one least-liked since it was perceived to be too complicated and tedious to use.

Similar to the results of expert users, graph-based tools outperformed the rest with respect to the overall satisfaction of the users with the tool (see Table 7.3: SUS score and Tool Rank). Although the ability to visualise the search space provided casual users (as well as expert users) an understanding of the available data and the possible ways of formulating queries, this was not the only reason which caused casual users to like these tools. The other major reason was the *visually-appealing* and *fun* (as described in the users feedback) interfaces of these tools. For instance, some of the repeated positive comments given by casual users for Affective Graphs are:

- *The interface is modern and visually appealing and made for a pleasant search experience.*
- *The animations and colours made it clear which concepts are connected and how.*

It is interesting to note that the visually-appealing interface not only provided users with a pleasant search experience but was also helpful (e.g. highlighting selected concepts) during query formulation. Similarly, the comment "*this is a nice, visual interface and fun to use*" was given by several users for Semantic Crystal.

Recall in Section 7.2.1, expert users preferred the approach of visualising the entire ontology (adopted by Semantic Crystal as shown in Figure 7.3a). This was indeed more appreciated by casual users, resulting in Semantic Crystal receiving higher SUS scores. Showing the whole ontology with the concepts and relations between them was very useful for casual users while formulating their queries. Surprisingly, the lack of this feature caused Affective Graphs to be perceived by casual users as the most complex and difficult to use, as shown in their feedback: 50% of the users found it to be: "*less intuitive and has higher learning curve than NL*".

Casual users prefer form-based approach

Another interesting finding of this study was that casual users most liked query approach is the form-based one (see Table 7.3: Query Language Rank). They perceived the form-based approach as a midpoint between NL- and graph-based approaches; it required less input time and they found it less complicated than the graph-based approach, while allowing more complex queries than the NL-based ones. Their feedback showed that they perceived the graph-based approach as laborious, especially for everyday use. They also stated that tools employing this approach are fun to use but also time-consuming, which is confirmed by the highest input time required by these tools (Affective Graphs and Semantic Crystal) as shown in Table 7.3. These users thus believed that the graph-based approach could be targeting users with specific information needs or certain use (e.g. infrequent complex queries). Interestingly, several users mentioned that being used to the free-NL approach adopted by traditional search engines (such as Google) may have biased their personal preference and they would be interested in spending more time using the view-based tools which could then change their perception.

Unexpectedly, casual users needed more attempts to formulate their queries with the form-based approach than with the graph-based one. From their feedback and our observations, it was found that the more attempts were due the presence of inverse relations often found between concepts in the ontology, which was viewed by the casual users as unnecessary redundancy. This led to confusion and thus required more trials to formulate the right queries. The geography ontology used in the evaluation contained several inverse relations such as *runsthrough* and *hasRiver*, connecting the concepts *State* and *River*. Therefore, for example, to query for the rivers running through a certain state, the two alternatives ("State, hasRiver, River" and "River, runsthrough, State") were adopted by users. Although the ontologies contain these relations and are therefore visualised by the tools in their interfaces, having two alternatives to formulate the same query was confusing for users. Casual users often stated that they did not know which alternative to choose or whether the two would provide the same answers. Tools ought to remove the burden from users and provide one unique way to formulate a single query.

Casual users liked controlled-NL support

Another major difference between expert users and casual users is related to the controlled-NL approach. As shown in Section 7.2.1, this approach was the least liked by expert users since it was deemed to be very frustrating as the restricted vocabulary limited their ability to express their information needs. Conversely, casual users found the guidance offered by suggesting query terms found in its vocabulary together with the prevention of invalid terms very helpful and highly supportive during query formulation. Interestingly, they preferred to be 'controlled' by the language model (allowing only valid queries) rather than having more expressiveness (provided by free-NL) that risked creating invalid queries. This provided them with more confidence in the queries they were building, since only correct queries are allowed and, therefore, the tool will have answers to return for them. This is supported by the casual users' feedback as the

most repeated positive comments for Ginseng (adopting the controlled-NL approach) were the following:

- *I liked that this tool allowed only correct queries.*
- *The suggestions and guidance to formulate queries was very helpful.*

Noticeably, casual users required the same average number of attempts (1.7) as expert users when formulating their queries with Ginseng (see *Number of Attempts* in Tables 7.1 and 7.3). The feedback given by the expert users showed that the low number was indeed due to frustration with the tool and an unwillingness to continue trying. However, for casual users this was in fact due to the guidance provided by the tool, which helped them complete their queries in a small number of attempts.

Discussion of the results of individual SUS questions

Again, this section presents the results of specific questions found in the SUS questionnaire which are focused on the usability and learnability aspects. Table 7.4 shows the results given by casual users for each tool for the following three questions:

- I thought the system was easy to use.
- I found the system very tedious / troublesome to use.
- I would imagine that most people would learn to use this system very quickly.

Table 7.4: Scores given by casual users for individual SUS questions. These questions are answered on a 5-point Likert scale ranging from *Strongly Disagree(1)* to *Strongly Agree(5)*. Bold values indicate best performing tool in that category.

Question (Strongly Disagree - Strongly Agree)	**AG**	**SC**	**K-S**	**Gins.**	**NLP-R**
System easy to use	2.9	**3.2**	2.9	3.1	2.7
System tedious to use	2.6	**2.3**	3	3	3.9
Learn to use the system quickly	2.9	3.2	3	**3.7**	2.8

The scores presented in Tables 7.4 and 7.3 show that SC – which obtained the highest overall SUS score and ranked as the most liked tool – managed to achieve the highest scores for the two usability questions. In contrast, Ginseng outperformed the rest of the tools (with high difference) in the learnability question. Indeed, casual users did not find it as difficult – to learn how to use – as the view-based tools (AG, SC and K-S). Additionally, their perception of NLP-Reduce, adopting free-NL, to be difficult to learn and use (supported by getting the lowest scores in the three questions) is due to the fact that its complete flexibility did not provide them with any support or guidance and left them not knowing how to formulate the right queries to answer their information needs.

Additionally, in line with the total SUS score, AG and Ginseng – which got nearly similar total SUS scores – alternated in their ranks in the usability questions. Interestingly, K-Search also managed to be in these same ranks (alternating with them) in

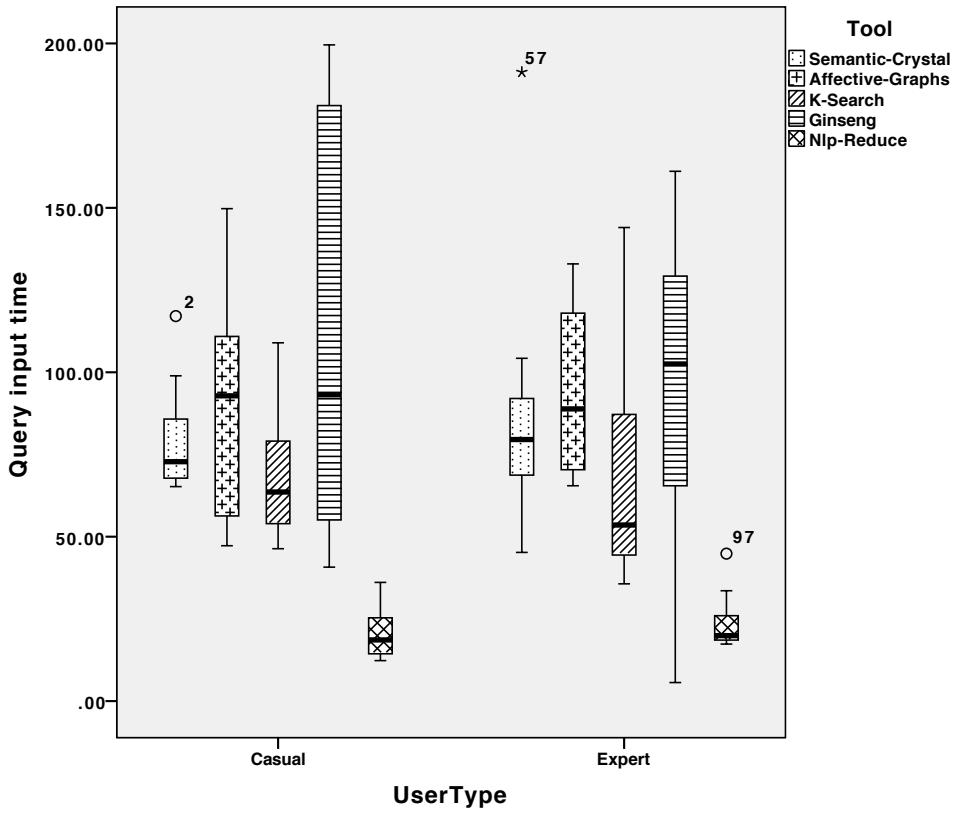


Figure 7.6: Time required by users to formulate their queries

all three questions. As discussed above, casual users liked the form-based approach and found it less complicated than the graph-based approach, which was shown in their feedback. However, the low SUS score obtained by K-Search was indeed affected by low scores of other individual questions such as '*I found the system unnecessarily complex*'. This was due to casual users having specific problems such as the 'inverse relations' issue, discussed earlier, and its appearance in the tree-like structure of K-Search which influenced the users' perception of its complexity.

7.2.3 Results Independent of User Type

This section discusses results and findings common to both types of users.

Free-NL simplest, fastest and most natural; suffer from habitability problem

On one hand, the free-NL approach was appreciated by both types of users for being simple, the most natural and the fastest to use (see Figure 7.6). Indeed, other approaches needed more time and effort since they required several mouse clicks, menu selections or a specific vocabulary to use while formulating a query. On the other hand, the results showed a frequent mismatch between users' query terms and the ones expected by the tool. This is caused by the abstraction of the search space and is known in literature as the *habitability problem* [KB10, p.2]. This is supported by the users' most

repeated negative comment: "*I have to guess the right words*". They found that they could get answers with specific query terms rather than others. For instance, using 'run through' with 'river' returned answers which were not given when using 'traverse'. Similar comments given by users included the following:

- *I need to know the language the system expects.*
- *It seems to use some sort of syntax which I do not know.*
- *It did not understand me, I did more than eight attempts for one question. I got frustrated by not getting answers back and not knowing what is wrong in my query.*

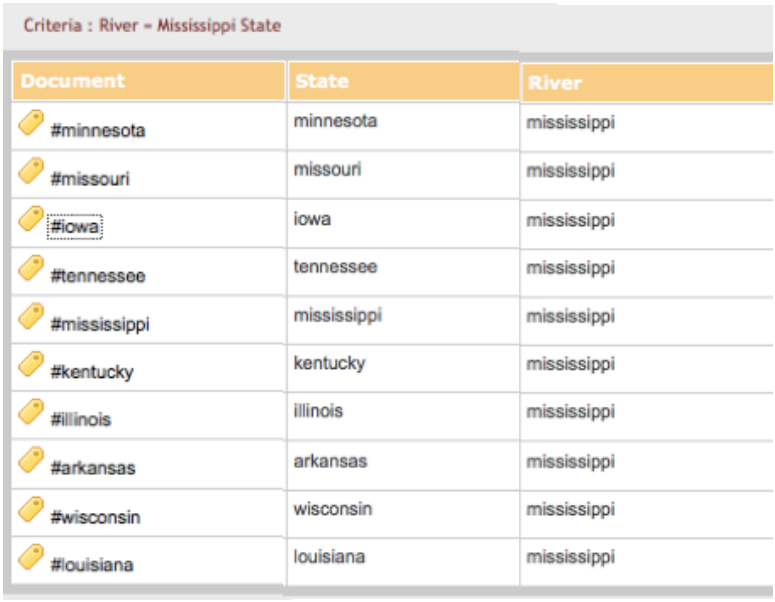
This is also confirmed by the tool adopting this approach (NLP-Reduce) getting the lowest success rate (20%), which shows that, on average, users could only answer around 20% of the questions. Although the exact performance is highly dependent on the tool and its underlying search techniques and employed algorithms, all NL-based tools – including the highest performing SOA tools – are faced with the habitability problem. Furthermore, requiring the highest *number of attempts* (4.1 and 4.2 by expert and casual users, respectively, as shown in Tables 7.1 and 7.3) support users' feedback that they had to rephrase their queries to find the combination of words the tool is expecting. Indeed, this is a general challenge facing natural language interfaces [LUSM11, KB07, ULL+07].

Form-based faster but more tedious than graph-based

Figure 7.6 shows that K-Search, which is employing a form-based approach, required both types of users less time to formulate their queries than the graph-based ones (approximate difference: 36% for experts and 14% for casuals). However, it was found to be more laborious to use than graphs especially when users had to inspect the concepts and properties (presented in a tree-like structure) to select the required ones for the query (see Figure 7.3c). This is a challenge acknowledged in the literature [LMUS07] for form-based approaches and is supported by the feedback given by users: the most repeated negative comment was "*It was hard to find what I was looking for once a number of items in the tree are expanded*". Additionally, this outcome suggests that input time cannot be used as the sole metric to inform usability of query approaches. Finally, it was found that visualising the search space in a graph-like structure made it easier for users to directly understand the relations found between the different concepts and how they can be connected in queries. Some users directly compared the graph- and form-based approaches in their feedback; for instance, the comment "*the ontology is not shown in a graph*" was repeated as a drawback for K-Search (employing form-based approach).

Results content and presentation affect usability and satisfaction

Besides performance and usability, it is important when evaluating semantic search tools to assess the usefulness of the information returned as well as how it is presented. In a separate study – not part of this thesis – my colleagues and I found that users have very high expectations of the usability and functionalities offered by a semantic search tool, especially with regards to the results management and presentation. In this



Criteria : River = Mississippi State

Document	State	River
#minnesota	minnesota	mississippi
#missouri	missouri	mississippi
#iowa	iowa	mississippi
#tennessee	tennessee	mississippi
#mississippi	mississippi	mississippi
#kentucky	kentucky	mississippi
#illinois	illinois	mississippi
#arkansas	arkansas	mississippi
#wisconsin	wisconsin	mississippi
#louisiana	louisiana	mississippi

Figure 7.7: Results returned by K-Search for the question "*What are the states through which the Mississippi runs?*"

evaluation, the sufficiency of the information presented and also the suitability of the presentation style were evaluated for each tool and compared to each other with respect to this aspect in the post-search questionnaires. These questions were: "*I liked the presentation of the answers*" and *The information given in the answers was sufficient*" as well as the ranking questions with respect to: 1) how much they found the results to be informative and sufficient, and 2) how much they liked the results presentation (e.g. readability, understandability, presentation style). The scores received by the tools for these questions are presented in Tables 7.1 and 7.3: *EQ1: liked presentation* and *EQ2: information sufficient.*

Within this context, our study found that the results presentation style employed by K-Search was the most liked by all users, as shown in Tables 7.1 and 7.3. It is interesting to note how small details such as organising answers in a table or having a visually-appealing display (adopted by K-Search) have a direct impact on results readability and clarity and, in turn, user satisfaction. This is shown from the most repeated comments given for K-Search:

- *I liked the way answers are displayed.*
- *The presentation format made it easy to interpret and understand the results.*

This is illustrated in Figure 7.7, which shows the answers returned by K-Search for the question "*What are the states through which the Mississippi runs?*".

Additionally, K-Search is the only tool that did not present a URI for an answer but

used a reference to the document using a NL label. This was favoured by users who often found URIs to be technical and more targeted towards domain experts. For instance, one user specifically mentioned having "*http://www.mooney.net/geo#tennesse2*" as an answer was not understandable. By examining the ontology, this was found to be the URI of *tennessee river* and it had the '2' at the end to differentiate it from *tennessee state*, which had the URI "*http://www.mooney.net/geo#tennesse*". This suggests that, unless users are very familiar with the data, presenting URIs alone is not very helpful.

By analysing users' feedback from the study mentioned above, we found that when returning answers to users, each result should be augmented with associated information to provide a 'richer' user experience. This was similarly shown by users' feedback in our study with the following comments regarding potential improvements often given for all of the tools:

- *Maybe a 'mouse over' function with the results that show more information.*
- *Perhaps related information with the results.*
- *Providing similar searches would have been helpful.*

Users often stated that such additional information would help them to better understand the results presented to them. They explained it as helping them in *putting the results within context*. For example, for a query requiring information about states, tools could go a step further and return extra information about each state – rather than only providing name and URI – such as the *capital, area, population* or *density*. Furthermore, they could augment the results with ones associated with related concepts which might be of interest to users [MMZ09, MBH+11]. Again, these could be instances of *lakes or mountains* (examples of concepts related to *state*) found in a state. This notion of relatedness or relevancy is clearly domain-dependent and is itself a research challenge. In this context, a notion of relatedness based on collaborative knowledge found in query logs is proposed in Section 11.2 – as part of future work.

Benefit of displaying generated formal query depends on user type

While casual users often perceived the formal query generated by a tool as confusing, expert users liked the ability to see the formal representation of their constructed query since it increased their confidence in what they were doing. Indeed, being able to perform direct changes to the formal query increased the expressiveness of the query language as perceived by expert users. Both of these features are provided only by Affective Graphs and Semantic Crystal.

Experts plan query formulation more than casuals

As shown in Table 7.5, with most of the tools expert users took more time to build their queries than casual ones. The feedback showed that the latter often spent more time planning and verbally describing their rationale (e.g. "so it understands abbreviations and it seems to work better with sentences than with keywords") during query formulation. Interestingly, studies on user search behaviour found similar results: Tabatabai and

Shore found that "*Novices were less patient and relied more on trial-and-error.*" [TS05, p.238] and Navarro-Prieto et al. showed that "*Experienced searchers ... planned in advance more than the novice participants*" [NPSR99, p.8].

Table 7.5: Query input time (in seconds) required by expert and casual users.

User Type	Affective Graphs	Semantic Crystal	K-Search	Ginseng	NLP-Reduce	p-value
Expert Users	88.86	79.55	53.54	102.52	19.90	0.001
Casual Users	72.8	75.76	63.59	93.13	18.6	0.001

7.3 Summary

This chapter has presented the usability study that was conducted to understand how expert and casual users perceived the usability of different query approaches. The study included evaluating five semantic search tools employing four query approaches: free-NL, controlled-NL, graph-based and form-based. Twenty subjects (10 expert users and 10 casual users) participated in the experiment, which followed a within-subjects design to allow direct comparison between the evaluated query approaches. Each subject was asked to perform five search tasks, of varying levels of complexity, querying Mooney geography dataset.

In order to assess the usability of the query approaches, I measured the efficiency, effectiveness and satisfaction of users with the evaluated tools. Hence, the data collected included: 1) the time required by users to formulate queries; 2) the number of attempts required for each query; 3) users' success rates in finding satisfying answers for the given queries; 4) users' input for two post-search questionnaires; and finally 5) post-experiment questions which required users to rank the tools according to specific criteria such as: *how much they liked the query approaches*. This data was then quantitatively and qualitatively analysed to assess usability and satisfaction. The study identified a number of findings, of which the most important are summarised below.

Graph-based approaches were perceived by expert users as intuitive, allowing them to formulate more complex queries. Casual users, despite finding these approaches difficult to use, enjoyed the visually-appealing interfaces which created an overall pleasant search experience. Showing the entire ontology helped users to understand the data and the possible ways of constructing queries. However, graph-based approach was judged as laborious and time consuming. In this context, the form-based approach required less input time. It was also perceived as a midpoint between NL-based and graph-based, allowing more complex queries than the first while being less complicated than the latter.

Additionally, casual users found the controlled-NL support to be very helpful, whereas expert users found it to be very restrictive and preferred the flexibility and expressiveness offered by free-NL. A major challenge for the latter was the mismatch between users' query terms and ones expected by the tool (habitability problem). Furthermore, the study showed that users often requested the search results to be augmented with

more information in order to have a better understanding of the answers. They also mentioned the need for a more user-friendly results presentation format. In this context, the most liked presentation was that employed by K-Search, providing results in a tabular format that was perceived as clear and visually-appealing.

To conclude, the usability study highlighted the advantage of visualising the search space offered by view-based query approaches. The findings suggest combining this with a NL-input feature that would balance difficulty and speed of query formulation. Based on these findings and conclusion, the graph-based approach (as the best performing approach) was selected for the user-based learnability study (discussed in Chapter 8) intended to investigate whether users' performance and efficiency in using the query approach would improve over time by training and frequency of use.

Chapter 8

Evaluating Learnability of a Graph-based Query Approach

As discussed earlier, the goal of the usability study presented in Chapter 7 was to understand users' requirements and needs in order to design a user-oriented query approach which balances both effectiveness and efficiency with usability. This study showed that both types of users liked the support given by view-based approaches (especially graph-based ones) in constructing queries through visualising the search space. However, these approaches were also found to require a fair amount of effort and time in constructing queries, which could affect their usefulness and users' efficiency while performing the intended search tasks.

Moreover, it is acknowledged that measuring usability in a one-time evaluation may not be sufficient for assessing user satisfaction with different query approaches, since the use of some systems employing these approaches is expected to require an amount of learning and therefore, assessing learnability as well would be essential. Based on this, the learnability study presented in this chapter is intended to understand if users' performance and perceived ease of use would change over time and frequency of use, in other words to investigate the learnability of view-based approaches.

The rest of this chapter is structured as follows: Section 8.1 provides an overview of the learnability study. In the same section, the methodology for the design of the study is described, with information covering the choice of the dataset and queries adopted, as well as the experiment setup. Section 8.2 describes the analysis performed on the data collected from the experiment, then presents the results of the study and discusses the most important findings.

8.1 Evaluation Design

As discussed in Section 4.4.1.6, learnability is an important criterion of usability that focuses on the ease of learning how to use a system or an interface. [Sha86] describes learnability as the relation of performance and efficiency to training and frequency of use. [Nie93] discusses how learnability can be measured in terms of the time required

for a user to be able to perform certain tasks successfully or reach a specified level of proficiency. Nielsen also notes that a system that is initially hard to learn could be eventually efficient [Nie93, p. 41]. The difference is, therefore, with respect to the time and effort required to reach a certain level of proficiency.

Nielsen's and Shackel's definitions of learnability were adopted in designing this evaluation to answer the following questions: 1) how easy (in terms of time and effort required) it is to learn how to use a semantic search query approach to answer a set of search tasks of different levels of complexity, 2) with training and frequency of use, what is the proficiency level users can achieve, compared to an expert's level as a benchmark. Additionally, I attempted to assess how easy it is for users to remember how to use the system which is sometimes included within the context of learnability [LP98].

As discussed in Section 6.3.2, the SWDF dataset (described below) was chosen for this evaluation. It was also explained how this influenced the choice of the recruited subjects, to be only expert users since casual users would not be familiar with the domain and could face difficulties understanding and reformulating the selected queries. Although it would have been beneficial to include casual users in this study, I believe that the results obtained with expert users only are still very useful in understanding the learnability of the graph-based approach, especially that it was easier to use and more liked by expert users than by casual users and therefore, it is not expected that casual users would perform better in such a study.

To answer the above research questions, ten expert users were asked to perform a given set of search tasks using Affective Graphs. The latter was selected from the set of tools adopting a view-based approach as a query mechanism for the following reasons (concluded from the usability study presented in the previous chapter): 1) it adopts a graph-based approach which was the most liked by expert users; 2) overall, it was the most liked tool by expert users and was perceived as '*good*' by casual users; and finally 3) it received the most positive feedback from both types of users. The experiment was conducted in three sessions (with different sets of tasks) over three consecutive days with the same users. Data from the experiments (such as query input time, number of attempts required for answering each query, and input of questionnaires) was recorded, and quantitatively and qualitatively analysed to answer the research questions.

8.1.1 Dataset and Questions

The Semantic Web Dog Food (SWDF) dataset, selected for this evaluation, contains information on publications, people, and organisations that were part of the main conferences and workshops in the area of Semantic Web such as WWW, ISWC and ESWC. At the time of writing this thesis, it contained information about 3858 papers, 9035 people, 2633 organisations, 31 conferences and 177 workshops with a total of 230569 unique triples[1].

The entities in the dataset are described using the *Semantic Web Conference Ontology (SWC)*. Additionally, the *FOAF* ontology is used for information about people and

[1] http://data.semanticweb.org/

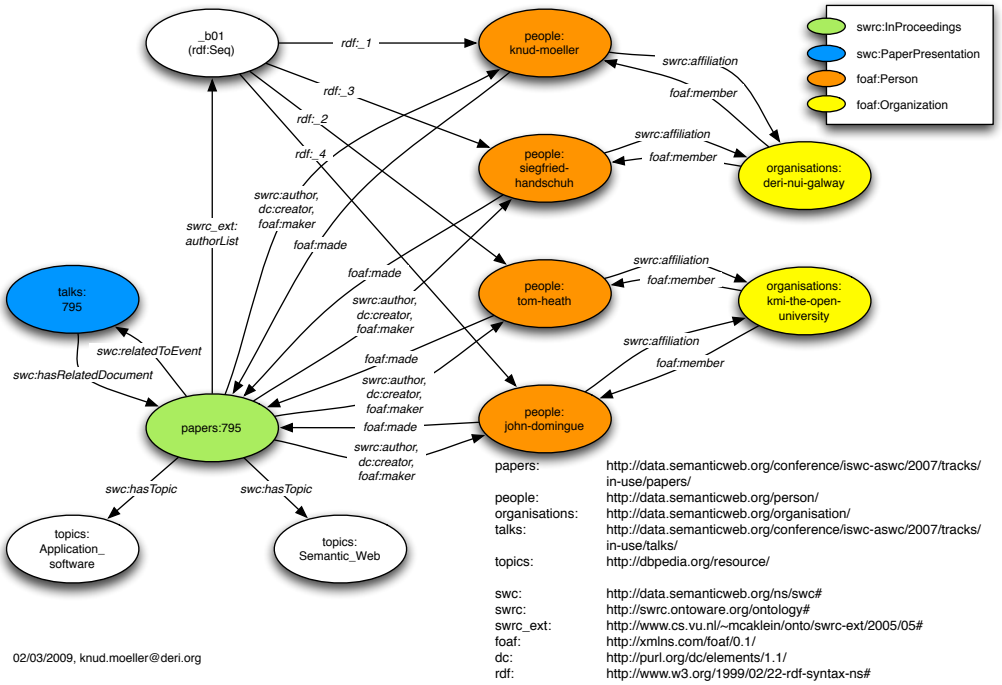


Figure 8.1: Example of how a paper, its authors, the corresponding talk and topics are linked together in the SWDF dataset

the *SWRC*[2] ontology is used – together with *SWC* – for information about all the entities including persons, organisations, publications (bibliographic metadata) and their relationships. Other ontologies used include *SIOC*, *Dublin Core* and *iCal*. Figure 8.1 shows an example of how a paper, its authors, the corresponding talk and topics are linked together in the SWDF dataset using the previously mentioned ontologies[3].

As described earlier, real-world queries were used in this study to conform to the representativeness, realism and reliability criteria. The (SPARQL) queries used in this study are provided by the USEWOD2011 data challenge[4]. First, the correctness of each SPARQL query was verified to exclude ones which contained errors. Then, the queries were analysed to understand the different types of requests made by users. The motivation for this analysis was that although task complexity and difficulty has received significant attention in the literature, neither IR evaluations – including large-scale initiatives such as TREC and INEX[5] – nor semantic search ones [HHM+10, WRE+10, BSdV10] considered these aspects in their design of evaluation queries. In these evaluations, queries were either selected from query logs or synthetically generated to simulate the first. In both approaches, attempts would be made to evaluate specific features or capabilities of the search tool or focus on a specific task type (such as fact finding or information gathering). Therefore, in this study, in addition to being real-world queries, they would be selected to cover different levels of complexity. These two criteria would help in selecting queries to be representative of the ones usually issued to semantic search

[2] http://ontoware.org/swrc/
[3] The example is taken from: http://data.semanticweb.org/documentation/user/faq
[4] http://data.semanticweb.org/usewod/2011/challenge.html
[5] http://www.informatik.uni-trier.de/~ley/db/conf/inex/index.html

systems. Indeed, [Nie93, p. 185] states that "the basic rule for test tasks is that they should be chosen to be as representative as possible of the uses to which the system will be eventually put in the field".

Following this analysis, four types of queries that are most often used were identified:

C = Concept, A = Attribute, F= Filter, R= Relation.

1. Simple Task (ST): $C_n A_n F_n$;
   $n = 1$
   Simple queries that comprise only one concept and one attribute but also a filter or a restriction value applied to the attribute. E.g. *Find the people with first name 'Knud'.*

2. Multiple Attributes Task (MAT): $C_n A_m$;
   $n = 1; m \geq 1$
   Increased number of attributes associated with only one concept, similar to depth search. E.g. *List the name, page and homepage of organisations.*

3. Multiple Concepts Task (MCT): $C_n R_m$;
   $n \geq 1, m \geq 1$
   Searching across multiple concepts, similar to breadth search. E.g. *List all the people who have given keynote talks.*

4. Complex Task (CT): $C_n A_m F_o R_p$;
   $n>1, m, o, p \geq 1$
   Include all the four components: concepts with relations linking them, attributes of the concepts as well as filters restricting the values of the attributes. E.g. *Find the page and homepage of each person whose status is 'Academia' and was a chair of a session event and find its location.*

Then, 12 (three from each category) of the most repeated queries (reoccurring queries with similar ontological concepts and properties but different instances) were selected. These queries were tested and validated on the basis of: 1) existence of results, and 2) conformance to the current versions of the ontologies used in the dataset. The final step was to manually translate the SPARQL queries into natural language (NL) – translation between NL and SPARQL is itself a challenge and beyond the scope of this work. The translation was based on the authors' understanding and interpretation of the information needs described by the original queries. The final set of queries is shown below:

Simple

1. Give me the people with first name 'Knud'.
2. Give me the inproceedings whose title contains 'Semantic Search'.
3. Give me the organisations whose name contains 'Karlsruhe'.

Multiple Attributes

1. List the name, page and homepage of organisations.
2. List the name, familyName and status of all people.
3. List the location, homepage and summary of all tutorial events.

Multiple Concepts

1. List all the conference venues and their meeting rooms.
2. List the programme committee members and the conference events they participated at.
3. List all the people who have given keynote talks.

Complex

1. Give me the description and summary of keynote talks which took place at 'WWW' conferences and the name of the presenter.
2. Give me the name, homepage and page of people who were workshop organisers for a workshop about 'Ontology Matching'.
3. Give me the page and homepage of each person whose status is 'Academia' and was a chair of a session event and give me its location.

8.1.2 Evaluation Setup

According to the requirements and design choices presented in Section 6.3.2 with respect to the number of subjects, ten subjects were recruited for this evaluation. These are expert subjects (2 females, 8 males), aged between 22–38 with a mean of 31 years [6]. The experiment took place in a controlled laboratory setting and subjects were rewarded for their time. They were drawn from the Organisations, Information and Knowledge (OAK) Group[7] within the Department of Computer Science at the University of Sheffield and from K-Now[8] – a software development firm doing research and working on semantic technologies. Note that these subjects are different from those who participated in the previous usability study, to avoid any influence on the results.

As discussed in Section 4.4.1.6, research on learnability was either focused on *initial learnability*: referring to the initial performance with the system, or *extended learnability*: referring to the change in performance over time [GFA09]. This study investigates the latter: *extended learnability*. Such studies are usually referred to as *longitudinal* studies, which are conducted over an extended period of time, with evaluation measures taken at fixed intervals, both of which determine the number of sessions required [Kel09]. It is thus important to decide on this period of time as well as the interval between the sessions. Similar to the choice of the number of subjects required for a usability study,

[6] This experiment is done in collaboration with SuvoDeep Mazumdar, the developer of Affective Graphs

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

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

the number of sessions presents a tradeoff between reliability of the evaluation (since it directly affects the amount of collected data and results), and its overhead. On the other hand, the interval between two evaluation sessions should be influenced by the expected/actual use of the evaluated system or interface. Hence, in order to strike a balance between reliability and overhead, it was decided to conduct the evaluation over three sessions. Additionally, since similar search tools are often used everyday, the three sessions took place over three consecutive days (with the same users), each of which took between 30-45 minutes. For the whole length of the evaluation, two experts were present for feedback purposes, which helped in identifying and recognising issues that would otherwise be ignored [GFA09].

At the beginning, subjects were introduced to the experiment and its goal, what is expected from them as well as any instructions required to be able to complete the experiment. Then for the first session, subjects were given hands-on training on how to use the interface to formulate queries (with examples of complete search tasks). After this, they were asked to formulate four questions in turn using the tool's interface. Then, subjects were asked to fill in two post-search questionnaires to capture their experience and level of satisfaction (*SUS* and *Extended* Questionnaires). For the second and third sessions, the same process was repeated. However, rather than training the users again, they were shown best practices of using the interface, and common difficulties that were highlighted during the first session were addressed. Additionally, they were given time (equal to the training time) to practice using the interface and do any kind of queries they were interested in. As shown in Section 8.1.1, 12 queries were chosen for this experiment. These were split into three sets, each containing four queries. To avoid any effects of the queries on the reliability of the experiment and the results, a query set was randomly chosen for each user and each session. At the end of the evaluation (after completing the three sessions), subjects were presented with a questionnaire to collect demographics data such as age, profession and knowledge of visual interfaces, among others. Finally, subjects were given the chance to provide any additional feedback they had about the evaluation.

The two ways proposed in literature to measure learnability are based on either using objective data to compare users' performance/efficiency over time or subjectively using learnability questions such as "*I found this interface easy to learn*". Similar to these studies and to allow for deeper analysis, both objective and subjective data covering the experiment results were collected. The first included: 1) *input time* required by users to formulate their queries, 2) *number of attempts*, capturing the average number of query reformulations required by users for a question, and 3) *success rate*, capturing the percentage of tasks successfully completed. The first two (input time and number of attempts) were used as the main metrics to measure users' efficiency. This data was collected using custom-written software which allowed each experiment session to be orchestrated. Additionally, subjective data was collected using two post-search questionnaires (presented at the end of each session as described above). The first was the SUS questionnaire (explained earlier in Section 4.4.3.3). The SUS questionnaire was included to understand whether usability of the interface as perceived by users is changed

by frequency of use, which also informs learnability. The second questionnaire (*Extended Questionnaire*) – also used to measure learnability – is one which was designed to include further questions related to the ease of use and learning of the interface as well as remembering how to use it. This questionnaire included five questions which are answered on a 5-point Likert scale (ranging from Strongly Agree/Easy to Strongly Disagree/Difficult) as shown below:

- The system's query language was easy to understand and use (Strongly Agree - Strongly Disagree).
- Tasks can be performed in a straightforward manner (Strongly Agree - Strongly Disagree).
- Exploring new features by trial and error is (Easy - Difficult).
- Remembering features and how to use them is (Easy - Difficult).
- Understanding the structure of the interface is (Easy - Difficult).

Furthermore, two open-ended questions were included to gather additional qualitative data after each session. These questions asked the subjects what they *liked* and *disliked* about the interface they tested. As discussed previously, these open-ended questions usually help researchers understand the users' rationale for answering closed-ended questions in a specific way. Additionally, for this study, analysing users' input for these questions and whether and how it changed from one session to another was useful in highlighting aspects related to users' perceptions and levels of satisfaction over time. For instance, the interface could be perceived as complex at the first session. However, if this changes over time, then it would be shown from the feedback given in the last session.

8.2 Results and Discussion

To quantitatively analyse the data collected, SPSS[9] was used, while for qualitative analysis, the open coding technique was used to categorise the data according to predefined aspects such as learnability and satisfaction.

Figure 8.2 shows the average time taken by users to formulate queries from all the four categories, while Figure 8.3 shows the same metric only for queries from the complex category. The results from the usability study presented earlier showed that users could formulate more complex queries with view-based approaches than with NL-based ones. Therefore, it was interesting to highlight and distinguish the results of queries in this category from the other categories. Additionally, as discussed in Section 4.4.1.6, one approach to assess the level of proficiency of users after a specific amount of training and use of the interface is to compare this level to experts' proficiency. Thus, two experts – in the Semantic Web field, with high knowledge of the interface (query approach) and the underlying data (domain) – were asked to formulate the same queries using the interface and their results were recorded. However, since experts did not require

[9] www.ibm.com/software/uk/analytics/spss/

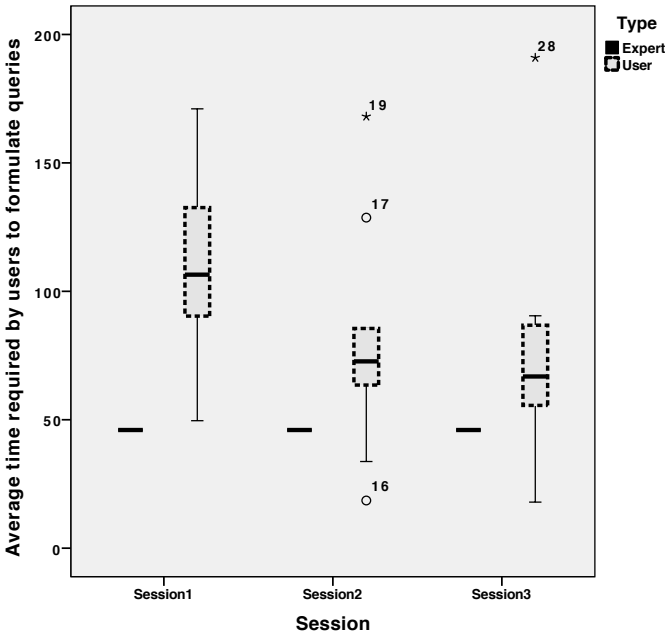


Figure 8.2: Input time required by users to formulate queries in the four categories

training and learning, in contrast to the users, only one session of the evaluation was conducted. Yet, in order to remove any effects or bias that could be introduced by selecting specific queries, the experts formulated all the 12 queries and the best results – across the queries as well as across the experts – were selected as the benchmark level. This level is shown in Figures 8.2 and 8.3 to facilitate the comparison.

The first finding summarised by both figures is that, in general, for all query types, the input time decreased over the evaluation period. Input time is a measure of efficiency, which is, in turn, used to measure learnability, and therefore, this finding supports the main hypothesis that users' efficiency improves with learning and frequent use of the interface. It is interesting to note how the significant amount of change in the time occurs between sessions one and two (Session 1: '106.3', Session 2: '71.4' and Session 3: '61.6'). This was also supported by our observations during the experiments which showed that users learnt much about the use of the interface in the first session which led to a high increase in their performance in the second session. The amount of learning however was not equally significant between the second and third sessions which influenced the amount of improvement. In contrast, for queries in the complex category, the improvement in the performance steadily continued over the three sessions (Session1: '132.5', Session2: '100.2', and Session3: '72.4'). By analysing users' feedback and also from our observations, we found that since complex queries were the most difficult for users to formulate, the learning continued after the second session together with the improved efficiency. Additionally, both figures show that after training and learning how to use the interface over the three sessions (amount of effort and time required), the users' proficiency level improved from '106.4' seconds (averaged over the four categories) to '61.6' seconds – compared to experts' level of '46' seconds – and from '132.5' seconds (for the complex category) to '72.4' seconds – compared to experts' level

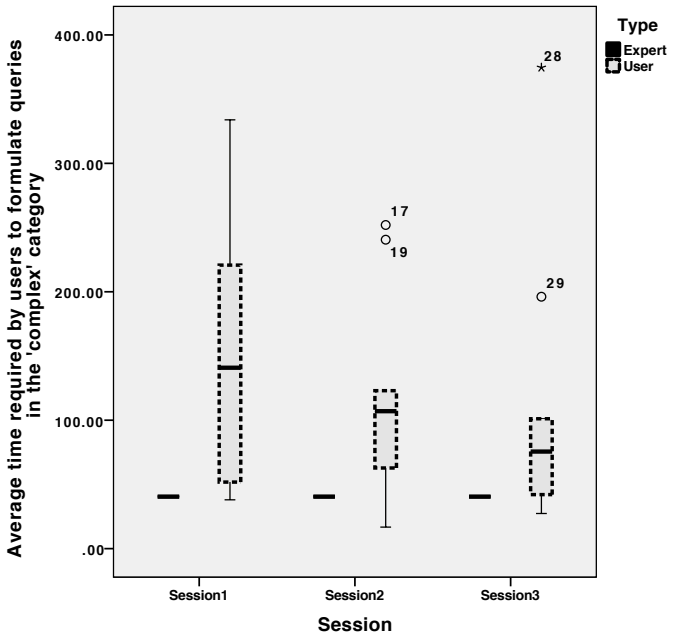


Figure 8.3: Input time required by users to formulate queries in the complex category

of '40.4' seconds.

The above findings which are based on objective data and measures are supported by the subjective data collected. Recall Section 8.1.2, the extended questionnaire consisted of five questions related to the ease of use and learning of the interface as well as remembering how to use it. Each question was answered on a 5-point Likert scale where '1' denotes 'Strongly Agree/Easy' and '5' denotes 'Strongly Disagree/Difficult'. Figure 8.4 shows the average score obtained for each of these questions in each session. The figure shows an increase in the scores given by the users for all the questions over the sessions which altogether informs learnability. The only exception is found in the scores of the question 'Remembering features and how to use them' since it was given the highest score in all sessions. This shows that users did not have difficulties remembering how to use the interface and its different features and functionalities.

Looking into more details, the two questions 'The system's query language was easy to understand and use' and 'Understanding the structure of the interface' are the main ones assessing usability of the interface. The scores of both questions are similarly improving between the first and the second sessions (query language: '2.5' then '2' and structure of the interface: '2' then '1.5') and then stabilising between the second and the third sessions (at 2 and 1.5 respectively). This is consistent with the above discussion in which we concluded that most of the learning was acquired between the first and the second sessions. Although the question 'Tasks can be performed in a straight-forward manner' is also related to the usability of the interface, our observations showed that the scores of this question were also influenced by the search behaviour. In other words, the scores were affected by the users' perception that they continuously (through the three sessions) adapted their search behaviour and learnt the 'ideal ways/best practices' to use the interface to answer their information needs. Therefore, as shown in Figure 8.4,

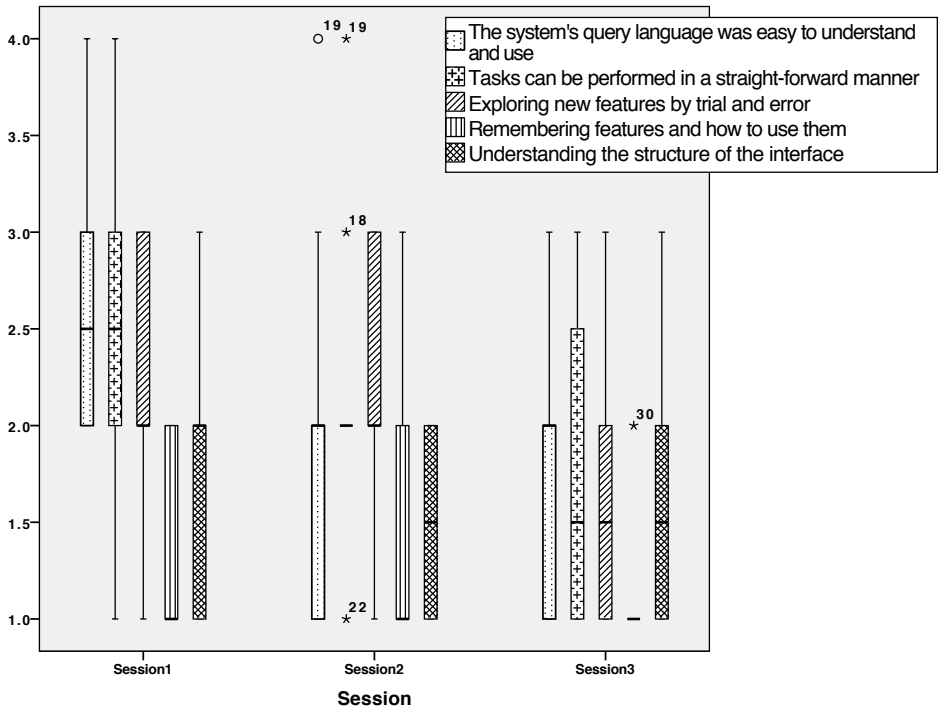


Figure 8.4: Scores for the questions from the *Extended Questionnaire*

the average score improved from '2.5' to '2' and then '1.5'.

The most repeated comments – informing learnability – given by the users in their answers to the open-ended questions are shown below:

- *My ability to use the system effectively and my confidence and speed in using the system grew over time.*
- *The system became easier to use and understand over time.*
- *I found satisfactory solutions to the questions much quicker in the second and third sessions than in the first.*
- *At first, the system seemed complicated to use. After using it a few times, I found it a lot easier to construct the queries.*

The feedback shows that practice and frequency of use of the system increased users' confidence and understanding of the query approach. This was also shown in the increase in the average SUS score given by the users over the three sessions of the experiment.

As shown in Figure 8.5, the average SUS score in the first session is appreciably high at 76.25 (good [BKM09]). Despite this high score obtained in the first session, users judged the system even higher: '82.5' (near excellent) in the next session. We believe that, as the users acclimatised to the system and the query mechanism, they tried different techniques to query and explore the data, providing them with more familiarity with the system's capabilities and limitations and eventually an understanding of the *ideal ways/best practices* to find answers for their information needs. The results of the third session showed a slight drop in the SUS score, from '82.5' to '81.25' (near excellent), which was still considerably higher than the score obtained in the first session ('76.25').

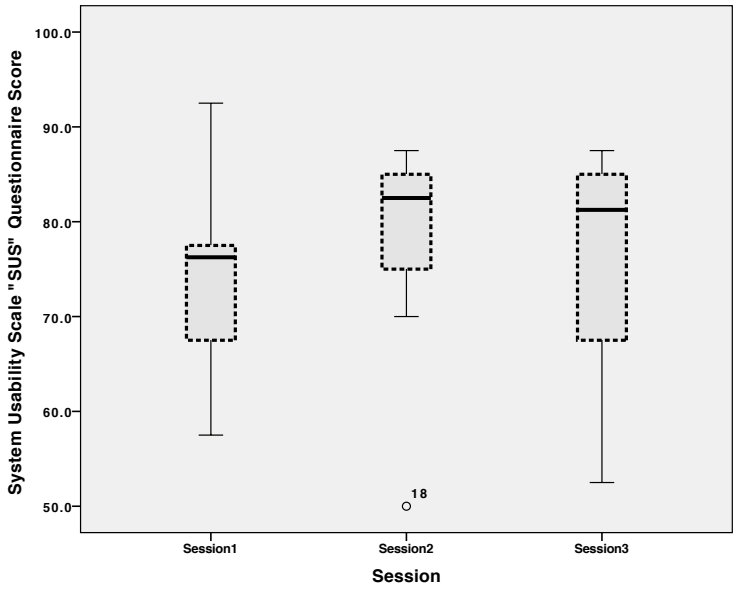


Figure 8.5: Average SUS score for the three sessions

Our understanding is that once the users adapted to the system, (most of the learning was acquired between the first and the second sessions as discussed earlier) their appreciation of the system, and the SUS score, increased. However, with more familiarisation obtained in the third session, users were less excited and their appreciation, and in-turn the SUS score, almost stabilised.

Similar to the usability study presented in the previous chapter, here I report the results of specific questions found in the SUS questionnaire which are focused on the usability and learnability aspects. Table 8.1 shows the results given by users for Affective Graphs in each session for the following three questions:

- I thought the system was easy to use.
- I found the system very tedious / troublesome to use.
- I would imagine that most people would learn to use this system very quickly.

Table 8.1: Scores given by users for individual SUS questions over the three sessions. These questions are answered on a 5-point Likert scale ranging from *Strongly Disagree(1)* to *Strongly Agree(5)*. Bold values indicate best performing session in that category.

Question (Strongly Disagree - Strongly Agree)	**Session 1**	**Session 2**	**Session 3**
SUS score	76.25	**82.5**	81.25
System easy to use	3.5	**3.9**	3.75
System tedious to use	2.1	**1.7**	1.8
Learn to use the system quickly	3.4	**3.6**	**3.6**

As shown in Table 8.1, the scores of the individual SUS questions in the three sessions show high correlation with the total SUS scores. In line with the discussion above, most

of the improvement can be seen in the scores of the questions between the first and the second sessions where most of the understanding of, and adaptation to, the system occurred. After the second session, more adaptation resulted in very slight changes to the scores, either dropping or improving. In contrast to the two usability questions, the scores given to the learnability question stabilized in the last two sessions, agreeing with the observations and feedback showing that almost all of the learning was acquired between the first and the second sessions, while changes between the second and the third session were most focused on users' search strategies and behavior – as discussed below.

8.2.1 Search Behaviour/Strategies

In addition to evaluating how users' performance changed over time, this study also allowed me to gain an insight into users' search behaviour and how they attempt to adapt it to identify more efficient search strategies which allow them to find answers for their information needs. To facilitate this, every attempt made by users to generate their queries was captured, including failed attempts. Users were observed during the sessions to note any change in the behaviour as well as feedback given with respect to the search strategies adopted during the information seeking process. One of the interesting findings of the analysis was the *increase in the average number of attempts (especially for complex queries) over the sessions* together with a *decrease in the amount of query input time.*

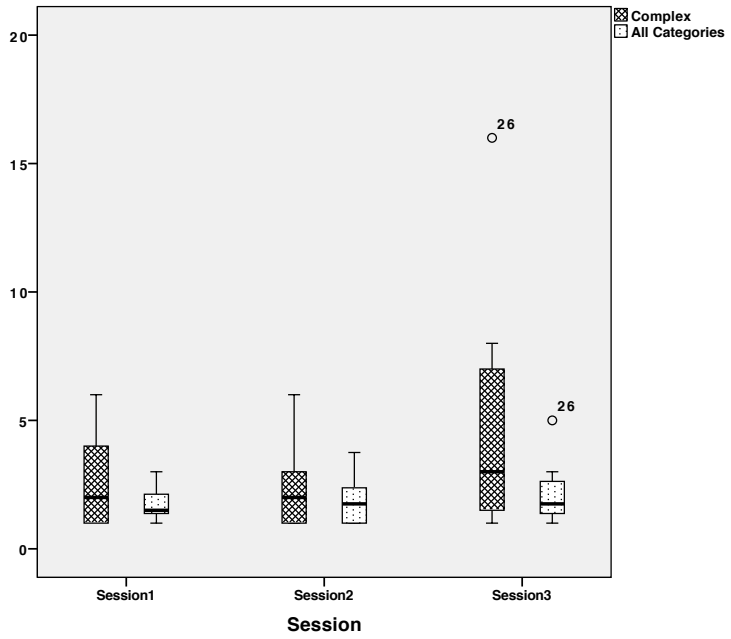


Figure 8.6: Number of attempts required by users to formulate queries

Figure 8.6 shows the average number of attempts required by users to formulate queries from all categories (*All Categories*) and from the complex category (*Complex*).

Observations during the first session indicated that, when they were new to the system, most users preferred to build large complete queries in a single attempt. This worked well for some queries, especially the simple ones. However, users seemed to experience difficulty while building queries for complex questions with this approach. Complex queries required users to connect multiple concepts using one or more relations, which often proved to be a challenging task, especially when users were acclimatising with the system and the visual query approach. This caused some frustration among users since they often resorted to clearing the page and building the entire query from the beginning. The training and the frequency of use of the system – during the later sessions – seemed to make users more comfortable with the visual query approach and more willing to examine different strategies to construct their queries. In the third session (where users were most familiar with the system), users' search strategies changed to building smaller queries using fewer concepts and relations and gradually building up on them. The overall search behaviour therefore evolved towards an approach where most users preferred performing short 'bursts' of queries as a means to examine the data as well as the validity of their (small) queries and thus progressively approaching a successful final query.

8.3 Summary

In this chapter, I have presented the learnability study that was conducted to understand if users' performance with a semantic search query approach would change over time and with frequency of use. The tool used for this study was *Affective Graphs* which was selected for being satisfying (*good* SUS rating) for both expert and casual users and for adopting a view-based interface which was shown to support users in constructing queries through visualising the search space. In this study, ten expert users were asked to perform 12 search tasks in three evaluation sessions (four different tasks in each session) which took place over three consecutive days. In order to assess the performance, objective data, such as query input time and number of attempts required for answering each task, was recorded. Additionally, users' search behaviour and strategies were observed throughout the evaluation sessions and finally, their experience was captured using questionnaires. The collected data was quantitatively and qualitatively analysed to assess learnability and satisfaction. Although the results showed an improvement in users' performance as well as satisfaction over time, it also showed that the effort and input time during query formulation, even after three practice sessions, could still be an issue for users with frequent search tasks.

This main finding together with the findings of the usability study presented in Chapter 7 motivated the design of a hybrid query approach, combining a graph-based approach with an NL-input feature. This hybrid approach would have the advantage of query formulation through visualisation of the search space (provided by the graph-based component) while being balanced in difficulty and speed (provided by the NL component). The details of this approach including its evaluation are discussed in Chapter 9.

Chapter 9

Hybrid Query Approach

9.1 Introduction

The results of the usability study presented in Chapter 7 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. Then, the learnability study presented in Chapter 8 investigated whether the effects of the latter could be alleviated by practice and frequency of use. The results showed an improvement in users' performance as well as satisfaction over time. However, it also showed that the effort and input time during query formulation, even after three practice sessions, could still be an issue for users with frequent search tasks. Therefore, my hypothesis, based on these findings, was that a hybrid approach which 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 would provide high level of support and satisfaction for users during query formulation. To evaluate this hypothesis, I developed a hybrid query approach – as a proof of concept – and conducted a third user-based study with expert and casual users to assess its usability and users' satisfaction.

The remainder of this chapter is structured as follows: related work of hybrid query approaches is presented in Section 9.2. The methodologies of the NL and the graph-based components are described in Sections 9.4 and 9.5. Then, the integrated hybrid approach is described in Section 9.3 together with a running example to illustrate its use. The evaluation of the approach and its results are presented in Section 9.6. Finally, conclusions and limitations of this work are discussed in Section 9.7.

9.2 Related Work

According to the classification presented in Section 3.3, a *hybrid query approach* uses a combination of approaches as a 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. [HC06] and [RSA04] 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, [BCC+08] used two query formats: keywords and forms to perform both keyword-based traditional search and semantic search, respectively, and combine the results of both. They defined *hybrid search* to be "the application of semantic (metadata-based) search for the parts of the user queries where metadata is available, and the application of keyword-based search for the parts not covered by metadata". Therefore, the two query approaches were separated and linked to two different underlying data indexes. The keyword-based approach was used to search traditional documents while the form-based approach was used to search semantic data and ontologies. Finally, [HV03] 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 facets, which facilitates multi-facet search[1]. [HV03] explain how the keywords search functionality is applied as follows: "The search keywords are matched against category names in the facets as well as text fields in the metadata. Then, a new dynamic view is created in the user interface. This view contains all categories whose name or other defined property value matches the keyword. Intuitively, these categories tell the different interpretations of the keyword, and by selecting one of them a semantically disambiguated choice can be made". This is similar to combining multiple query approaches but also combining semantic search techniques to improve the results of traditional search, as described above.

In a similar way, the hybrid approach presented here combines two different query approaches (NL and graph-based) to support users during query formulation. It attempts to benefit from the strengths of the graph-based approach in visualising the search space, while trying to balance the effort required during query formulation using a NL input feature.

9.3 NL-Graphs: Putting the Hybrid Approach into Practice

As discussed earlier, the hybrid approach presented here[2] 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. The intuition is to conform to the results and conclusions drawn from the usability studies discussed in Chapters 7 and 8, that is to benefit from the strengths of the graph-based approach in visualising the search space, while trying to balance the effort required during query formulation using a NL input feature. Additionally, based on the design choices discussed in Section 6.3.1, NL-Graphs features the following main components,

[1] [HV03] use *facets* interchangeably with *views* and *multi-facet* with *view-based*

[2] The integration of the NL component with Affective Graphs was done in collaboration with Suvodeep Mazumdar, the developer of the latter.

shown in Figure 9.1:

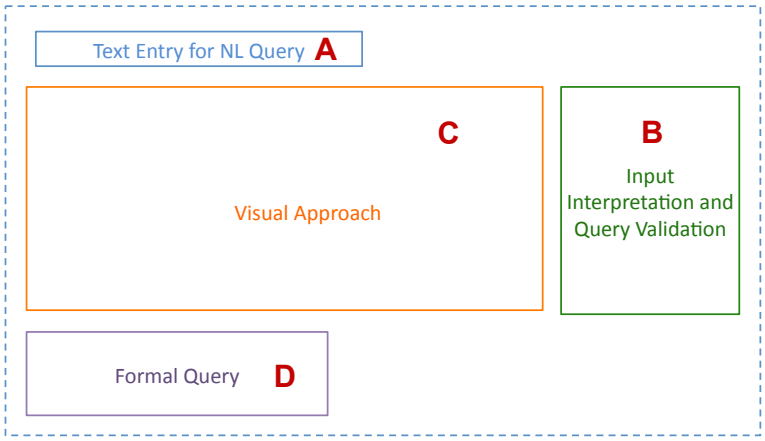


Figure 9.1: A Mockup of *NL-Graphs*.

- 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): As discussed in Section 9.4 and throughout the thesis, the main difficulty 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 SOA 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 visual query which is automatically built using the NL-component's output – as will be explained below; 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 Chapter 7. The results showed that the formal representation of the constructed queries provided experts

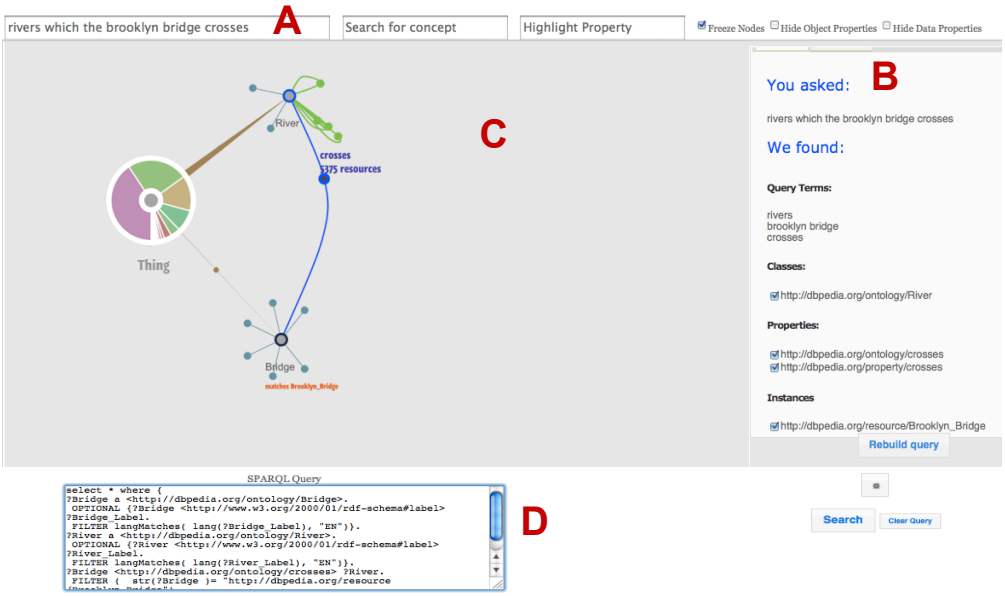


Figure 9.2: NL-Graphs interface for the query "*rivers which the brooklyn bridge crosses*".

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 Chapter 7). 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.

9.3.1 NL-Graphs Architecture

Implementing the above requirements resulted in the Web interface shown in Figure 9.2. As shown in the workflow presented in Figure 9.3, a user's query is firstly 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, which are explained below in Section 9.4 are applied in order. The output of these steps is a set of candidate triples as shown in the example below[3]:

```
<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 with an ontology concept or with an instance, these mappings are similarly passed to the graph-based

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

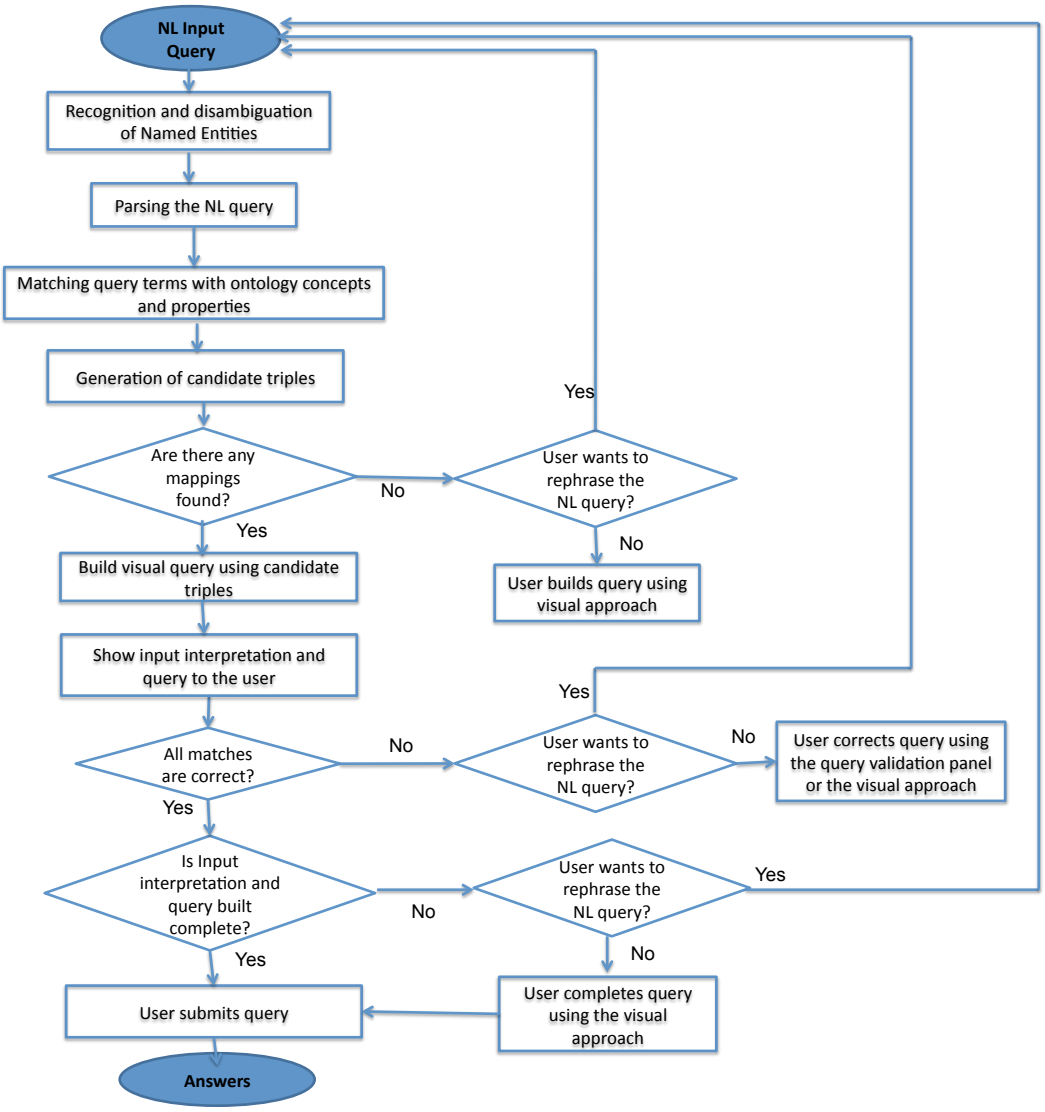


Figure 9.3: NL-Graphs workflow

component to be visualised in the graphical panel (Figure 9.2: C). This is performed within the next step in the workflow: "*Build visual query using candidate triples*", as explained below.

In the graph-based component, any concepts found – in the list of terms received from the NL-component – are analysed first. Each unexplored concept is loaded, along with all its respective data and object properties. The instances are then analysed where each instance's 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, a final stage of *Rationalisation* occurs in which the visual query is loaded, the query variables are inspected and the formal query is completed.

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 9.2: 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 9.2: 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.

9.3.2 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 9.2:(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` 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 correct interpretation and complete query built and therefore, continues to submit the query to get direct answers as shown in Figure 9.4.

Presentation of results is a challenging research problem which can have different

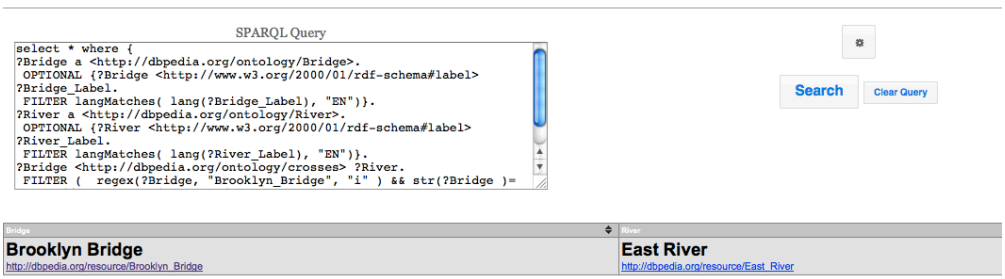


Figure 9.4: NL-Graphs results for the query "*rivers which the brooklyn bridge crosses*".

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 as shown earlier in Chapter 7. However, since this is not the focus of this work, we decided to present results – direct answers – in a simple and clear format, 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?*". As will be illustrated in Section 9.4, since no exact match is identified for the query term *founded*, then the algorithm returns all matches whose similarity exceeds a predefined threshold (as explained below, the threshold is set to 0.791, which was shown by [dSSOH07] to be the best value). Therefore, the properties `dbo:foundingYear, dbo:foundingDate, dbo:foundedBy, dbp:founder` and `dbp:foundation` are generated as candidate mappings and presented in the validation panel, as shown in Figure 9.5.

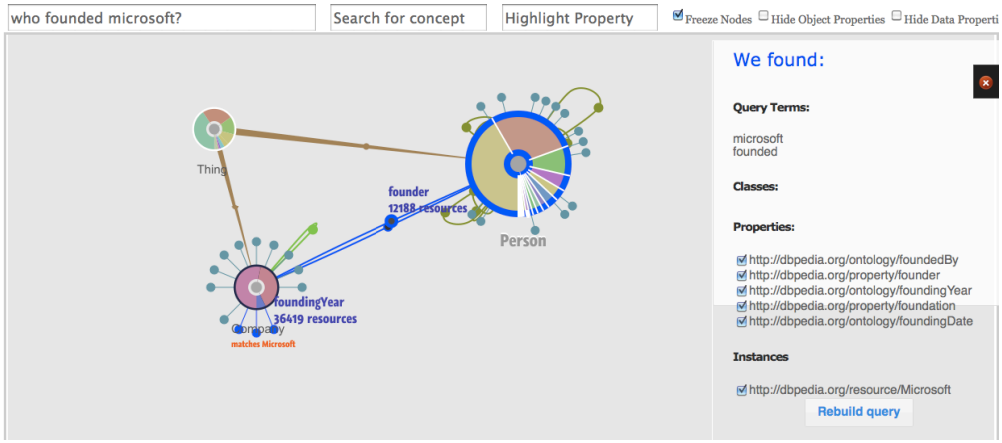


Figure 9.5: NL-Graphs input interpretation for the query "*who founded microsoft?*".

Additionally, the data properties `dbo:foundingYear, dbo:foundingDate` and `dbp:foundation` 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 in the validation panel and choose to *Rebuild Query*. Both panels are then updated to reflect these changes, as shown in Figure 9.6. 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

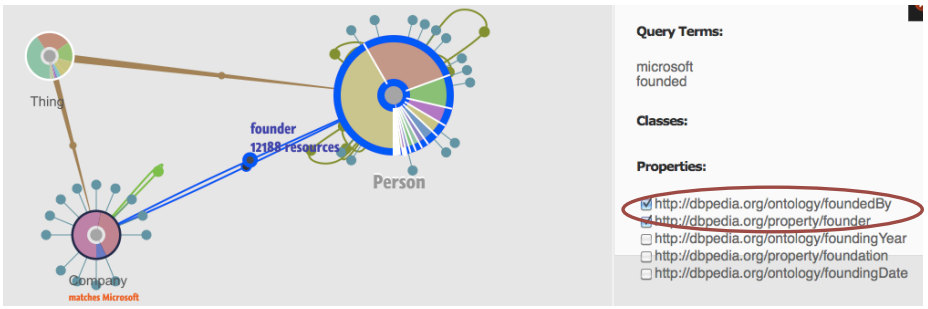


Figure 9.6: A user validates and corrects the input interpretation of NL-Graphs for the query "*who founded microsoft?*".

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 9.7 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.

Figure 9.7: NL-Graphs input interpretation for the query "*brooklyn bridge traverse which river*".

9.4 The Natural Language Component

As discussed in Section 3.3.3, systems adopting a NL approach employ different syntactic as well as semantic parsing techniques in the process of understanding the natural language query and generating the corresponding formal query. This process starts by annotating the query to identify different word forms (usually performed using a standardised parser such as *Stanford Parser*). While some systems only use the output of the POS tagger (adopted in TBSL [UBL+12]), others may depend on the complete parse

tree generated by the parser (adopted in FREyA [DAC10]).

Additionally, named entities (NEs) found in a query are recognised in this step by mapping proper nouns (annotated by the Parser) to resources in the ontology (adopted in PowerAqua [LMU06]) or in a separate step using a named entity recogniser, or using a combination of both techniques (adopted in QAKiS [CAC+12]). As earlier noted, the computational cost of matching NEs to resources in one or more ontology can increase in proportion to the ontology size. Furthermore, NEs can refer to multiple real world entities thus necessitating disambiguation. This is either performed using the context of the query and the structure of the ontology or using a disambiguation technique (provided by the NER or developed for this purpose).

Then, the second step is to map the identified word forms (nouns, verbs, etc.) to ontological terms. Most of the time, this is done using the structure of the ontology and the POS tags. For instance, nouns are mapped to both classes and properties while verbs are only mapped to properties (used in TBSL). However, two difficulties usually arise in this step. The first is *semantic ambiguities* arising due to *polysemy* (single word with more than one meaning) and affects precision of results by providing false matches. The second is *missing matches* arising due to *synonymy* (multiple words with the same meaning) and affects recall by causing true (semantic) matches to be missed.

To tackle the first difficulty, one strategy is to use the query context together with the ontology structure to identify the correct sense of the polysemous word (employed in PowerAqua). Another strategy is to engage the user in clarifying the ambiguity faced by the system (employed in Querix). Finally, to have the best of both worlds, some systems seek help from the user only if they failed to automatically resolve the ambiguity (employed in Freya). To tackle the second difficulty, *query expansion* – adding words to the query which are related in some sense to query terms – is usually adopted. The different sources for gathering these related query terms were discussed in Section 3.4.3.1. An additional difficulty arises when query expansion is attempted for a *polysemous* word. For example, in order to answer the question "How tall is ...?", the query term *tall* needs to be mapped to the ontology property *height* (a term related to tall). However, the term *tall* is also polysemous and has different senses including (from WordNet):

- "great in vertical dimension; high in stature; tall people; tall buildings, etc."
- "too improbable to admit of belief; a tall story"
- "impressively difficult; a tall order"

Therefore, the term must be disambiguated and the right sense identified (the first in this example), before attempting to gather related terms. For instance, [LMU06] uses a disambiguation approach inspired by [MSS03] in which a specific WordNet synset is considered relevant only if one of its senses (separate words in a WordNet synset) exists in the synonyms, hypernyms, hyponyms, holonyms or meronyms of an ancestor or a descendant of the synset. Others consider all senses of a polysemous word and use their related terms for query expansion [WUCB12], a strategy which could increase noise and irrelevant matches and, therefore, affect recall.

Finally, ranking the ontological terms generated from the previous step could be

required to identify the most relevant mappings for a specific query term. For example, the question "What are the official languages of the Philippines?" can generate many mappings for the terms *official* and *language* including `dbo:Language`, `dbp:officialLanguages`, and `dbp:languages`[4]. Ranking is usually performed based on a set of syntactic and semantic similarity algorithms. Then, depending on whether the preference is for precision or recall, only the best match, the ones whose similarity exceed a certain threshold or all of the matches are then selected.

As illustrated in the architecture of NL-Graphs, presented in Section 9.3.1, the output of the NL-component is passed to the graph-based component which provides a visual representation of the query for the user and generates the equivalent formal query. Therefore, the input, task, and output of the NL-component are as follows:

- *Input*: Free-form natural language query: keywords, phrases or full questions.
- *Task*: Syntactic and semantic parsing of the input query.
- *Output*: Candidate triples of ontological terms (concepts, properties and instances) and relations between them.

For example, the input query "*Which television shows were created by Walt Disney?*" generates the output:

```
?television_show <dbo:creator> <res:Walt_Disney>.
?television_show <dbp:creator> <res:Walt_Disney>.
?television_show <dbo:creativeDirector> <res:Walt_Disney>.
```

To perform the *task* and, indeed, based on the above review, some of the most commonly adopted techniques – in high-performance SOA such as Freya, PowerAqua and QAKiS – are followed in each step. Therefore, as will be illustrated in the following sections, Stanford parser is used to parse the NL query. NEs are recognised using AlchemyAPI[5] which had the best NE recognition performance as shown in [RT11]. A set of advanced string similarity algorithms and ontology-based heuristics are used to match query terms to ontology concepts and properties. Finally, a high performance WSD approach has been developed specifically for use within the NL-component (since no standardised modules or services were available to perform this task with high performance [RT11]).

The novelty of the approach described above lies in the combination of a template-based approach for understanding users' queries – in an attempt to capture the context around the word and reduce the possibility of a wrong match as a result of a word-based match – with performing query expansion and WSD using BabelNet as a wide-coverage knowledge base.

Since the WSD is a separate module which is used in these steps as will be discussed below, its design and evaluation are first presented in the next section. Then,

[4]The prefix `dbo` refers to: `<http://dbpedia.org/ontology/>`; the prefix `dbp` refers to: `<http://dbpedia.org/property/>`

[5]http://www.alchemyapi.com

Section 9.4.2 presents the details of the above steps and Section 9.4.3 presents the evaluation of the approach using the dataset provided by the *Question Answering over Linked Data (QALD-2)* workshop[6].

9.4.1 Word Sense Disambiguation (WSD)

WSD approaches are either supervised, unsupervised or knowledge-based. Supervised approaches require a large amount of sense-annotated examples, which are usually hard to obtain [IV98, BP02, CDBB09], especially in an *all-words* scenario (in which the task requires disambiguating every word in a text). Unsupervised approaches depend on unannotated corpora which are usually automatically extracted with high levels of noise – for instance, through web search [CDBB09] – and, therefore, are known to suffer from low performance [NLH07]. Finally, knowledge-based approaches use dictionaries and lexicons such as WordNet to perform WSD and are often considered a middle ground between the other two approaches. Widely used knowledge-based approaches include Lesk-like [Les86] as well as graph-based approaches. Graph-based approaches – unlike the others – attempt to find globally optimal solutions by analysing the whole graph which contains words and their senses (as nodes) and relations (as edges) connecting them. Although these approaches have been gaining more attention recently for their high performance [NP12, ADLS09], I used an extended-Lesk approach for the following reasons: 1) it was one of the highest performing in knowledge-based approaches (see UPV-WSD in [NLH07]); [PN10] showed only 2% difference when compared with a graph-based approach, and 2) it is a simpler approach to test my hypothesis of improving the mapping between NL queries and LD ontologies using a WSD approach with high-coverage knowledgebase (BabelNet). Note that, at the time of implementing this WSD algorithm, the WSD provided within BabelNet was not yet available for use, and therefore, a specifically designed WSD was implemented to perform this task.

WordNet is the predominant resource used in such knowledge-based WSD approaches; however, it has been argued that its fine granularity is the main problem for achieving high performance in WSD [IV98, NP12]. In light of this, I adopted a knowledge-based approach which uses the alternative BabelNet[7] [NP12] for disambiguation. BabelNet is a very large multilingual ontology with wide-coverage obtained from the automatic integration of WordNet and Wikipedia; in addition, it has been enriched with automatic translations of its concepts.

Additionally, the evaluations conducted and presented earlier show that users more frequently use short queries with keywords or phrases as opposed to full sentences, do not follow correct grammar rules and randomly construct their queries (not following a specific order for the query terms). For example, for a query requesting information about states that run through the Mississippi river, observed user queries included the following:

- "Mississippi river states"
- "Mississippi states"

[6] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge&q=2
[7] http://babelnet.org

- "states with Mississippi"
- "which states Mississippi run through"
- "states run Mississippi river"

Therefore, it was decided to apply a WSD approach which considers the input sentence as a bag of words with no differentiation between them. The approach is based on the Lesk approach while extending it with different lexical and semantic relations. Since contradicting results have been reported on the value of including specific relations (e.g. using the hyponymy relation [BP03, FMS03]), I conducted an analysis on the use of different relations and their effect on the performance. Then, based on the results, a set of features are selected and adopted in my proposed approach.

9.4.1.1 Disambiguation Algorithm

In broad terms, WSD uses sources of knowledge to collect information about the context in which the target word appeared and also about its different meanings. The target word is then disambiguated by comparing the context information (referred to as context vector or context bag) with each sense's information (referred to here as synset bag) and selecting the one with the maximum overlap. To disambiguate a polysemous target word w_t:

1. Construct the context bag C. To do this, the Stanford parser [KM03] is used to parse the input sentence. For each word w tagged as a noun, verb, adjective, or adverb, and excluding the target word, do the following:
   (a) Retrieve all different synsets for w – only associated with its part of speech (POS) – from the knowledge base.
   (b) For each synset s_i, construct synset bag S_i according to the specified relations (discussed in Section 9.4.1.3).

   Then, aggregate all the synsets' bags $S_{1..i}$ to form C.
2. For the target word w_t, retrieve all its different synsets – only associated with its POS – from the knowledge base. For each synset s_i, do the following:
   (a) Construct synset bag S_i according to the specified features.
   (b) Calculate the overlap score between C and S_i.
   $Score = 2 * |C \cap S_i|/(|C| + |S_i|)$
   The Dice coefficient [Dic45] is used to normalise the number of overlapping words in the two bags by the size of the bags.

   The 'winning' synset bag is the one with the highest overlap score; if a tie occurs, one is selected at random from the set of tied synset bags.

9.4.1.2 Relations Used

In BabelNet, the information added from a Wikipedia page (W) mapped to a specific WordNet synset includes:

Table 9.1: Precision (P), Recall (R) and F-Measure (F_1) results of applying different features to the WSD approach.

Feature	**P**	**R**	**F_1**
Baseline	58.09	57.98	58.03
Syn	59.14	59.03	59.09
Syn + hypo (level 1)	62.16	62.07	62.12
Syn + gloss examples (WN)	61.97	61.86	61.92
Syn + gloss examples (Wiki)	61.14	61.02	61.08
Syn + gloss examples (WN + Wiki)	60.21	60.10	60.16
Syn + hyper (level 2)	60.36	60.26	60.31
Syn + semRel	59.65	59.54	59.59
Syn + hypo + gloss(WN)	64.92	64.81	64.86
Syn + hypo + gloss(WN) + hyper	65.28	65.18	65.23
Syn + hypo + gloss(WN) + hyper + semRel	65.45	65.33	65.39
Syn+hypo+gloss(WN)+hyper+semRel+relGlosses	**69.76**	**69.66**	**69.71**

1. labels; e.g. given the page *Play (theatre)*, the words *play* and *theatre* are added;
2. set of pages redirecting to W, e.g. *Playlet* redirects to *Play (theatre)*;
3. set of pages linked from W, e.g. links in the page *Play (theatre)* include literature, comedy, etc [NP12].

This information is referred to as *wikipage information*. Therefore, for a WordNet synset S and its associated Wikipedia page W, our reference to a relation/feature such as *synonyms* of this synset will always mean: "WordNet synonyms of S in addition to lemmas of *wikipedia information* of W". Similarly, hyponyms would refer to "WordNet hyponyms of S in addition to lemmas of *wikipedia information* of each wikipedia page associated with each hyponym synset", and so forth for the other relations. In addition to *synonyms* and *hyponyms*, I include *hypernyms*, *glosses* in addition to *attribute*, *see also* and *similar to* which are semantic relations defined by WordNet [MBF+90].

9.4.1.3 Evaluation and Discussion

Table 9.1 lists the precision, recall and f-measure achieved by our approach (on the SemEval-2007 coarse-grained all-words dataset[8]) when the context and synset bags are extended using the listed features (in addition to the synonyms). The baseline is based on disambiguation using random sense assignment.

Extending the context bags with hyponyms of the synsets provided a large increase in performance ($\approx$ 4%). Only direct hyponyms (referred to as *level 1* in Table 9.1) were considered; it was found, empirically, that as the number of hyponyms is usually much higher than that of hypernyms (which is often only one or a few), the direct level is sufficient to provide an increase in performance (indeed, adding more levels tended to negatively affect it). For hypernyms, the direct level did not give sufficient information and therefore, only a modest increase in performance (+1.22%) was gained when hypernyms were added up to the second level (consistent with [BP03]).

[8] http://lcl.uniroma1.it/coarse-grained-aw/index.html

Adding glosses of synsets provided the next largest increase in performance. I only include the examples part of a gloss; preliminary experiments showed that examples provide the best performance, followed by the whole gloss and then the definitions part, with a significant difference between the examples and the definition part of around 2.2% – a similar finding to that reported by [BRM04]. In addition to the WordNet gloss, the BabelNet gloss includes the lemmas of the first sentence found in the wikipedia page associated with the WordNet synset [NP12]. Although the latter could provide useful information for some synsets, my observations showed that, on average, it caused more noise and therefore, negatively affected the precision. In Table 9.1, "WN" refers to WordNet gloss, "Wiki" to the first sentence from Wikipedia, and "WN + Wiki" to both. Finally, the semantic relations: *attribute*, *similar to* and *see also* (referred to in Table 9.1 as *semRel*) provided around 0.5% increase in performance. After examining the effect of each feature separately, I proceeded by combining them, one by one, in order of their contribution to the performance. Adding hyponyms, glosses, hypernyms and semantic relations raised the performance by approximately 7.3%. Following [BP03] who noticed an improvement in performance by adding glosses of related synsets, I also examined adding glosses of synsets related to the main synset through one of the hyponyms, hypernyms, or semantic relations (referred to as *relGlosses* in Table 9.1) to the query and synsets' bags. This caused an additional large improvement in performance ($\approx$ +4.3%) to reach an f-measure of 69.71%.

After this step, adding more features to the disambiguation approach caused the performance to either stabilise or start decreasing, similarly noted by [VLL04], especially as the context gets longer. The effect of adding more features varied with respect to the sentence size: the performance improved further when applied to short sentences; indeed, its average performance was negatively affected by longer sentences. For sentences where the number of keywords was less than 7 (100 sentences), the approach achieved an f-measure of 81.34% (the results in Table 9.1 include sentences with more than 15 keywords, thus reducing the overall f-measure).

It is worth noting that the queries commonly used in the evaluation of semantic search approaches (e.g. [TM01] and QALD challenge[9]) tend to contain no more than five keywords. Therefore, I believe this can be considered as the final performance of the algorithm.

9.4.2 Sense-aware Search

Users exhibit a general preference for short NL queries, consisting of keywords or phrases, as opposed to full sentences and a random query structure with no specific order for the query terms [RLME05] (and based on an analysis of user queries used in the evaluations discussed in the previous chapters). This section describes the approach adopted to process free-form natural language queries and try to establish the underlying 'meaning' of the query terms (using word sense disambiguation; see Section 9.4.1) allowing them to be more accurately associated with the underlying dataset's concepts and properties.

[9] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/

This approach consists of four stages:

1. Recognition and disambiguation of Named Entities.
2. Parsing the NL query.
3. Matching query terms with ontology concepts and properties.
4. Generation of candidate triples.

9.4.2.1 Recognition and Disambiguation of Named Entities

Named entities are recognised using AlchemyAPI[10] which had the best NE recognition performance in a recent evaluation of SOA recognisers [RT11]. However, AlchemyAPI exhibits poor disambiguation performance [RT11]; in this work, each NE is disambiguated using the algorithm described in Section 9.4.1.1. For example, for the question "*In which country does the Nile start?*", the term *Nile* has different matches in BabelNet. These matches include:

- http://dbpedia.org/resource/Nile_(singer)
- http://dbpedia.org/resource/Nile_(TV_series)
- http://dbpedia.org/resource/Nile_(band)
- http://dbpedia.org/resource/Nile

Although one could select the last URI as an exact match to the query term, syntactic matching alone can not guarantee the intended meaning of the term, which is better identified using the query context. Using our WSD approach, the correct match for *Nile*, as a river, would be selected since more overlapping terms are found between this sense and the query (such as *geography, area, culture* and *continent*) than the other senses.

9.4.2.2 Parsing and Disambiguation of the Natural Language Query

The second step is to parse the NL query, which is done using the Stanford parser [KM03]. However, since users are not expected to adhere to correct grammar or structure in their queries, the approach does not make use of the generated parse trees but only use lemmatisation and part of speech (POS) tagging. Each query term is stored with its lemma and POS tag except for previously recognised NEs which are not lemmatised. Additionally, the position of each term with respect to the rest of the query is identified and used in the later steps. For example, the question "*Which software has been developed by organisations founded in California?*" from the QALD-2 dataset generates the following outcome:

- software: at position `1` and POS `NP`
- developed: at position `2` and POS `VBN`
- organisations: at position `3` and POS `NNS`
- founded: at position `4` and POS `VBN`
- California: at position `5` and POS `NP`

[10] http://www.alchemyapi.com

Equivalent output is also generated when using keywords or phrases. At the end of this step, any proper nouns identified by the parser, and which were not recognised by AlchemyAPI as NEs, are disambiguated as described in Section 9.4.1 and added to the set of recognised entities. This ensures that, for the example used above: "*In which country does the Nile start?*" the algorithm does not miss the entity *Nile* because it was not recognised by AlchemyAPI.

9.4.2.3 Matching Query Terms with Ontology Concepts and Properties

The terms generated from the above step are then matched to concepts and properties in the ontologies being used. Noun phrases, nouns and adjectives are matched with both concepts and properties, while verbs are only matched with properties. After gathering all candidate ontology matches that are syntactically similar to a query term, these are then ordered using two string similarity algorithms: *Jaro-Winkler* [Win90] and *Double Metaphone* [Phi00]. Jaro-Winkler depends on comparing the number and order of common characters. Similar to Monge Elkan [ME96] which is used by [DAC10], it gives a high score to terms which are parts of each other. This is useful since ontology concepts and properties are usually named in this way: for instance, the term *population* and the property *totalPopulation* are given a high similarity score using this algorithm. An additional advantage of this algorithm is efficiency; [CRF03] found it to be an order of magnitude faster than Monge-Elkan. The threshold for accepting a match is set to 0.791, which was shown by [dSSOH07] to be the best threshold value. Double Metaphone captures words based on a phonetic basis and is, therefore, useful to capture similarly sounding terms.

If a query term produces no matches, its lemma is used for matching. If no matches were found, derivationally related forms of the query term are then used. For example, the property *creator* in the question "*Which television shows were created by Walt Disney?*" is only found after getting these forms for the term *create*.

After this, if no matches are found, the query term is then disambiguated using the WSD algorithm described in Section 9.4.1.1 and terms related to the identified synset are gathered. These terms are used to find matches in the ontology, based on both their level in the taxonomy (the nearest, the better) and in order of their contribution to the WSD as shown by the results in Section 9.4.1.3. Thus, synonyms are used first, then semantic relations (the appropriate ones), followed by hyponyms, and finally hypernyms. For nouns, no semantic relations are used, while for verbs, *see also* is used and finally, for adjectives, *attribute* and *similar to* are used in that order. Indeed, the attribute relation is very useful for adjectives since, for example, the property *height* is identified as an attribute for the adjective *tall*, which allows answering the question "How tall is ...?". The query term is marked as *not found* if no matches were found after all expansion terms have been used. Note that superlatives and comparatives are not matched to ontology concepts or properties; they are used in the next step to generate the appropriate triples.

9.4.2.4 Generation of Candidate Triples

After all terms have gone through the matching process, the query can be interpreted in terms of a set of ontology concepts, properties and instances that need to be linked together. The structure of the ontology (taxonomy of classes and domain and range information of properties) in addition to BabelNet are used in this step, as will be explained next.

Three-Terms Rule

Firstly, each three consecutive terms are matched (using the information about their relative positions as explained in Section 9.4.2.2) against a set of templates. These templates are the result of an empirical analysis of a wide range of queries gathered from semantic search evaluations ([TM01] and QALD challenge[11]). The intuition behind this step is to find a subject with a specified relation to a given object. Then, the ontology matches associated with each term are used to generate one or more candidate triples. For instance, the question "*Which television shows were created by Walt Disney?*" which can also be given as keywords *television show, create, Walt Disney* matches the template *concept-property-instance* and generates the following triples:

```
?television_show <dbo:creator> <res:Walt_Disney>.
?television_show <dbp:creator> <res:Walt_Disney>.
?television_show <dbo:creativeDirector> <res:Walt_Disney>.
```

Triples generated from the same query term are ordered according to the similarity of the matches found in them with respect to this term. In this example, the two properties `dbo:creator` and `dbp:creator` are ordered before `dbo:creativeDirector` since they have a higher similarity score with the query term *create*. Similar questions that would be matched to this template include *airports located in California*, and *actors born in Germany*. The other templates capture the different ordering that can be found in the query such as *instance-property-concept* in the question "*Was Natalie Portman born in the United States?*" or *property-concept-instance* in the question "*birthdays of actors of television show Charmed*". Note that in the last example, since the type of the instance *Charmed* is identified as 'television show', the latter is excluded during triples generation making it: *birthdays of actors of Charmed*.

Two-Terms Rule

Some user queries contain fewer than three pieces of information, thus preventing the application of the *Three-Terms Rule*. This can happen in three situations:

1. There is no match between the derived terms and any three-term template.
2. The template did not generate candidate triples.
3. There are fewer than three derived terms.

[11] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/

For example, the second situations occur in the second part of the question "*In which films directed by Garry Marshall was Julia Roberts starring?*" in which the terms *Garry Marshall*, *Julia Roberts* and *starring* would be matched to an existing template but without generating candidate triples. The requirement that the domain of the property (in this case: `Film`) must be the type of one of the instances was not met.

For the above scenarios, the same process of template matching and triples generation for each pair of consecutive terms is followed. For instance, the question "*area code of Berlin*" generates the triples:

```
<res:Berlin> <dbp:areaCode> ?area_code.
<res:Berlin> <dbo:areaCode> ?area_code.
```

Comparatives

As explained earlier, superlatives and comparatives are not matched to ontology terms but used here to generate the appropriate triples. For comparatives, there are four different scenarios that were found from an analysis of the queries in datasets used by different semantic search evalutions (e.g. Mooney dataset [TM01] and datasets used in QALD challenges[12]). The first is when a comparative is used with a numeric datatype property such as the property `numberOfEmployees` in the question phrase "*more than 500000 employees*". This information is known from the range of the property. In this case the following triples are generated:

```
?company <dbp:numEmployees> ?employee.
?company <dbp:numberOfEmployees> ?employee.
```

These triples are ordered according to their similarity to the the original query term (*employee*) and a choice is made between using the best match or all matches depending on the priority of the algorithm (i.e., whether to favour precision or recall). The chosen triples are then added to the following ones:

```
?company a <dbo:Company>.
  FILTER ( ?employee > 500000)
```

The second scenario is when a comparative is used with a concept as in the example *places with more than 2 caves*. Here, the approach would generate the same triples that would be generated for *places with caves* to which the aggregate restriction: `GROUP BY ?place HAVING (COUNT(?cave) > 2)` would be added.

In the third scenario, the comparative is used with an object property which, similarly, requires an aggregate restriction. In the example *countries with more than 2 official languages*, the following restriction is added to the normal triples generated between country and official language.

```
GROUP BY ?country HAVING (COUNT(?official_language) > 2)
```

[12] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/

The fourth and most challenging scenario can be illustrated by the question "*Which mountains are higher than the Nanga Parbat?*". The difficulty here is to identify the property referred to by the comparative term (which is 'elevation' in this example) to get its value for the given instance and then do a comparison with this value. While [DAC10] tackles this challenge by generating suggestions using the datatype properties associated with the concept and asking the user for assistance, this can be an overhead on the user. Our algorithm tries to select the best relation according to the query context. Firstly, all numeric datatype properties associated with the query concept (in this case *mountain*) are identified. These are: `latS`, `longD`, `prominence`, `firstAscent`, `elevationM`, `latD`, `elevation`, `longM`, `latM`, `prominenceM`, and `longS`. Using our WSD approach, each of these properties is first disambiguated to identify the synset which is most relevant to the query context. Then, the selected synsets of all the properties are put together and treated as different meanings of a polysemous word in order to have the WSD approach identify the most related synset to the query. Using this technique, the algorithm correctly selects the property `elevation` to use and then proceeds to find mountains with elevation higher than that of the instance *Nanga Parbat*. In order to verify whether the WSD algorithm was affected by the abbreviations (such as `latM`), the same question was asked after replacing the abbreviations by their equivalent word (*latitude* for `latM`). The fortunate result was that it still selected `elevation` as the most relevant property since it had more overlapping terms with the query than the others.

Indeed, it is more challenging to identify this link between the term and the appropriate property for more generic comparatives like *larger* in the query *cities larger than Cairo*. Several interpretations arise, including area of the city, its population or density. The ability to resolve this scenario is future work.

Superlatives

For superlatives, two different scenarios were identified. Either it is used with a numeric datatype property such as in the example *city with largest population*, or with a concept as in *what is the highest mountain*. In the first scenario, the normal triples between the concept `city` and property `population` are generated, in addition to an `ORDER BY` clause together with a `LIMIT` to return the first result.

The second scenario is more challenging and similar to the last comparative scenario explained above and is indeed tackled using the same technique. All numeric datatype properties of the concept are identified and the most relevant one (identified by our WSD approach) is used in the query. Again, in this example, the term *highest* is successfully mapped to the property `elevation`.

9.4.2.5 Integration of Triples and Generation of SPARQL Queries

As discussed earlier, the output of the NL-component is the set of candidate triples (generated from the previous step) which are passed to the graph-based component. However, in order to evaluate the approach and compare it with SOA, a final step is included to generate the equivalent SPARQL query by integrating the candidate triples.

Information about the query term position is used to order the sets of triples originating from different query terms. Furthermore, for triples originating from the same query term, care is taken to ensure they are executed in the appropriate order until an answer is found (when higher precision is preferred and thus not all matches are used). For example, in the question "*Which software has been developed by organisations founded in California?*", the terms in the first part – *software, developed, organisations* – generate the following triples:

```
?software <dbp:developer> ?organisation.
?software a <dbo:Software>.
?organisation a <dbo:Organisation>.
```

And the terms in the second part — *organisations, founded, California* — generate the following triples:

```
?organisation <dbp:foundation> <res:California>.
?organisation a <dbo:Organisation>.
```

To produce the final query, duplicates are removed while merging the triples and the `SELECT` and `WHERE` clauses are added in addition to any aggregate restrictions or solution modifiers required.

9.4.3 Evaluation

This section presents a comparative evaluation of our approach using the DBpedia training[13] dataset provided by the 2nd Open Challenge on Question Answering over Linked Data (presented in Section 4.5.3). Results were produced by the QALD-2 evaluation tool[14].

9.4.3.1 Results

Table 9.2 shows the performance in terms of precision, recall and f-measure, in addition to coverage (number of answered questions, out of 100) and the number of correct answers (defined as P=R=(F_1)=1.0). Our approach (SenseAware) is compared with QALD-2 participants [UCL+12]: SemSeK, Alexandria, MHE and QAKiS, in addition to BELA [WUCB12], which was evaluated after QALD-2 but using the same dataset and questions.

The results show SenseAware is very promising especially in terms of correctness: 76% of answers were 'correct'. It also achieves higher performance than the other approaches except for BELA. The latter has higher values for P, R (and F_1) since it favours these over coverage and correctness (only 31 answered of which 55% were 'correct'). After excluding out-of-scope questions (as defined by the organisers) and any containing references to concepts and properties in ontologies other than DBpedia since they are not yet indexed by our approach, we had 75 questions left. The 21 questions – out of 75 – that our approach couldn't provide an answer for fall into the following categories:

[13] The test data was not available at the time of this experiment.

[14] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=evaltool&q=2

Table 9.2: Results for our approach (SenseAware, shown in bold) with SOA approaches.

Approach	**Answered**	**Correct**	**P**	**R**	**F_1**
BELA	31	17	0.62	0.73	0.67
QAKiS	35	11	0.39	0.37	0.38
Alexandria	25	5	0.43	0.46	0.45
SenseAware	**54**	**41**	**0.51**	**0.53**	**0.52**
SemSeK	80	32	0.44	0.48	0.46
MHE	97	30	0.36	0.4	0.38

1. No matches were found for one or more query terms after query expansion.
2. Matches were found for all query terms but question type is out-of-scope.
3. Question requires higher level of reasoning than is currently provided.

Examples of the first category are: *What did Bruce Carver die from?*, in which the term *die* should be mapped to the property `deathcause` and *Who owns Aldi?* in which the term *owns* should be mapped to the property `keyPerson`. Questions that are not yet addressed are mainly the ones which require identifying the right property to use depending on the answer type. An example is *When was the Battle of Gettysburg?* which requires using the property `date`. Another example is *In which films did Julia Roberts as well as Richard Gere play?*. Here, our approach could not relate the concept `films` with *Richard Gere*. Although, it is designed to maximise the chance of linking concepts, properties and instances in the query, without being affected with the structure of the sentence, this version cannot yet link a term (*films*) that is being successfully related to other terms (*Julia Roberts*) to an additional term (Richard Gere). However, it can still solve complex questions that require relating terms that are not consecutively positioned (e.g. *films* and *Julia Roberts*) in the question "*In which films directed by Garry Marshall was Julia Roberts starring?*". Finally, examples of questions in the third category are *Is Frank Herbert still alive?* which requires understanding that the expression *still alive* means not finding a value for the death date of the person.

9.4.3.2 Discussion

In designing an approach to answer user questions, it is usually difficult to decide whether it is better to favour precision or recall, since it is well known that an increase in one commonly causes a decrease in the other. In fact, which to favour depends not only on the users but on their specific information need at some point. This was experienced while designing my approach since I had to decide on the following choices to be in favour of precision or recall:

Query Relaxation

Consider the question "*Give me all actors starring in Last Action Hero*". This question explicitly defines the type of entities requested as *actors* which justifies querying the dataset for only this type. Hence, the triple: `?actor a <dbo:Actor>` would be added to restrict the results generated from: `res:Last_Action_Hero dbp:starring ?actor` to only these who are actors. However, the current quality of Linked Data would be a major problem with this choice, since not all entities are typed [NGPC12], let alone typed cor-

rectly. This causes the previous query to fail, and only succeeds to return the required answer after query relaxation, i.e., removing the restricting triple `?actor a <dbo:Actor>`. This choice is in favour of *recall*. It affects precision since, for the question "*How many films did Leonardo DiCaprio star in?*", following this technique would also return answers that are *TV series* rather than *films* such as `res:Parenthood_(1990_TV_series)`. Our decision was to favour precision and keep the restriction whenever it is explicitly specified in the user's query.

Best or All Matches

The decision to use only the best match found in the ontology for a query term or all matches whose similarity exceeds a certain threshold can directly affect precision and recall. For instance, the term *founded* in the question "*software developed by organisations founded in California*" has several matches including `foundation` and `foundationPlace`. Using only the best match (`foundation`) would not generate all the results and, in turn, affects the recall. On the other hand, if these properties were not relevant to the query, this would harm the precision. To balance both precision and recall, our algorithm uses all matches while employing a high value for the similarity threshold and performing several checks against the ontology structure to assure relevant matches are only used in the final query.

Query Expansion

When a query term is not found in the ontology, query expansion is performed to identify related terms and repeat the matching process using these terms. However, in some scenarios, this expansion might be useful to increase the recall, when the query term is not sufficient to return all the answers. Therefore, it would be useful to perform the expansion for all query terms even if they had matches in the ontology. An example of this is when one of the two terms *website* or *homepage* are used in a query and both of them have matches in the ontology. Using only one of them could affect recall for some queries. On the other hand, the quality/relevance of expansion terms (for polysemous words) depends fully on the WSD approach. If a wrong sense was identified for a query term, this list will be noisy and lead to false matches. Additionally, the disambiguation process is computationally expensive and therefore, for these reasons, query expansion is performed only when no matches are found in the ontology for a term or when no results are generated using the identified matches.

9.5 The Graph-based Component

As discussed earlier, the intuition behind adding a graph-based component to the hybrid approach was to benefit from its strengths in visualising the search space and supporting users in formulating their queries, especially complex ones. Therefore, Affective Graphs – the most liked tool adopting a view-based approach in the usability study presented earlier – was selected as the graph-based component in our hybrid approach. The rest of this section provides a brief overview of the design methodology and architecture of

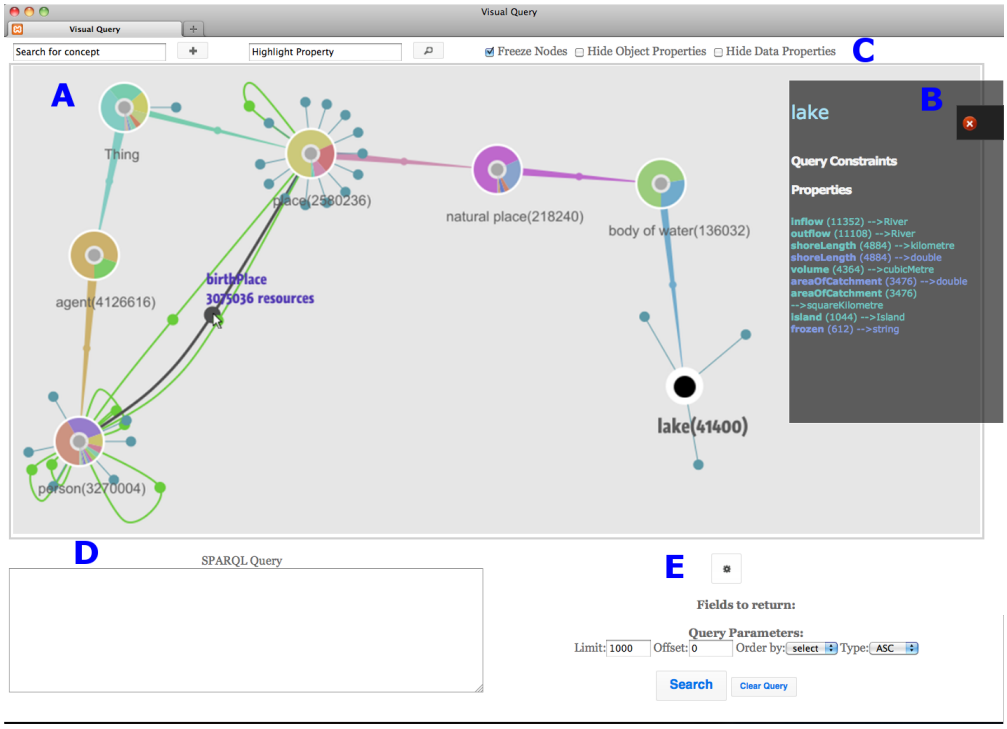


Figure 9.8: A screenshot of *Affective Graphs*, where the node currently on focus is 'Lake'. Section A contains the interactive node-link representation of the data, Section B contains contextual information relevant to the concept currently being explored (here, Lake), Section C contains search elements and controls the visual rendering of the node-link graph, Section D shows the SPARQL query being generated for search and Section E contains advanced features to modify the query.

Affective Graphs [SDE+13].

Semantic Data is highly graphical in nature, where concepts and classes are linked to each other with relations. The kind of relations depict how we conceptualise such data - this is the focus of our graph-based approach. Affective Graphs was developed as a highly interactive and graphical system which uses visual means to communicate semantic information to users. The starting point of Affective Graphs is the rationale that directly abstracting semantic data leads to a node-link representation. A study of the literature revealed several visual analytic and aesthetic techniques and principles which were employed in designing Affective Graphs. Affective Graphs is a web-based tool that employs a client-server mechanism to query Linked Data endpoints on the basis of their interactions with end users.

Figure 9.8 shows a screenshot of Affective Graphs, exploring information about Lakes in DBpedia. The interface consists of five components: a main graphical visualisation interface that presents a node-link graph (A); a contextual information display window (B); search boxes and visualisation control panel (C); a SPARQL query display (D); and an advanced control panel (E). The interface presents the underlying ontology as a node-link graph, where nodes represent concepts and links represent properties. Each

node is rendered as a circular object, embedded with a pie chart. The pie chart indicates a distribution of the number of instances of the subclasses within the respective concept. This helps the user understand the data content and also how it is structured.

We identified two major types of properties : a `typeOf` or `subclassOf` hierarchical property; and a non-hierarchical property. Since Affective Graphs employs a node-link representation, we stress on distinguishing the two types of properties. Hierarchical properties are represented as triangles, where the base of the triangle lies closer to the parent, and the apex lies closer to the child. Non-hierarchical properties are represented as bezier curves connecting the object and subject classes.

Users can click on different sections of a pie chart to "expand" their search to the subclass. This triggers the creation of another node, with a hierarchical property connecting the previous node to the new one. The new node being created is also provided with a pie chart illustrating the distribution of the subclasses of its concept. Other queries are also triggered which fetch the properties related to the newly created node, and any properties discovered are rendered as a non-hierarchical curve connecting the new node to other open nodes. The nodes are positioned using a customised force-directed layout, which only executes after a new node is generated. This enables the force direction to quickly select the best position for the new node, but also allows the user to reposition the node where it is decided that it fits best.

Right clicking on the links and the nodes displays a pop-up context menu with Affective Graphs items such as adding the object to the query, configuring a constraint or toggling visibility for nodes. Once the concepts of interest have been explored, the context menu can be used to build a specific query. Right-clicking a property selects it and adds its subject and object to the query. For example, in the Figure, adding birthPlace to a query will create the following query triple:

```
?person dbprop:brightPlace ?place
```

The query triple will then be added to the present query, and the complete formal query will be displayed in the SPARQL query box.

9.6 Evaluation

It is important to note that the NL-component and the graph-based component (Affective Graphs) were evaluated separately in terms of their performance; and usability and learnability, respectively. Information covering these evaluations can be found in Section 9.4 and in [SDE+13], respectively. Therefore, the rest of this section is focused on the evaluation conducted to assess the usability of the hybrid approach (implemented in NL-Graphs) as a new query mechanism. Additionally, note that the current version of NL-Graphs has been tested with DBpedia. However, 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.

Recall, the hypothesis which motivated the idea of the hybrid approach presented above was that the latter would benefit from the strength of the graph-based approach in visualising the search space, while balancing the time and effort required during query formulation using a NL input feature. To evaluate this hypothesis, 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. The study involved 24 subjects (12 expert users and 12 casual users) who were asked to answer a set of search tasks using NL-Graphs' interface.

In order to assess the efficiency, effectiveness and usability of the approach as well as users' satisfaction, both objective and subjective data were collected. The first included the time required by users to formulate queries, the number of attempts required for each query as well as the number and reasons for failures – if occurred – in answering the search tasks. On the other hand, subjective data was collected using post-search questionnaires, test leaders' observations, in addition to screen recordings capturing the interaction of users with the interface.

9.6.1 Dataset and Questions

As discussed in Section 6.3.2, DBpedia was selected as the dataset for this evaluation. DBpedia 3.8, the version used in this study, consists of 1.89 billion triples while the ontology covers more than 500 classes which form a subsumption hierarchy and are described by more than 2000 different properties[15]. In addition, to allow assessing the usefulness of the hybrid approach, queries with which NL-based approaches would face problems while attempting to answer were selected. These problems would be, for instance, resulting from the difficulty of mapping user query terms to ontological ones or understanding complex questions such as those containing comparatives, superlatives or advanced constraints. Based on the evaluation and analysis presented earlier in Section 9.4.3, five queries were selected from the DBpedia training and test data provided by the 2nd Open Challenge (QALD-2)[16]. 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:foundingDate`, `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

[15] http://dbpedia.org/Ontology
[16] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=challenge&q=2

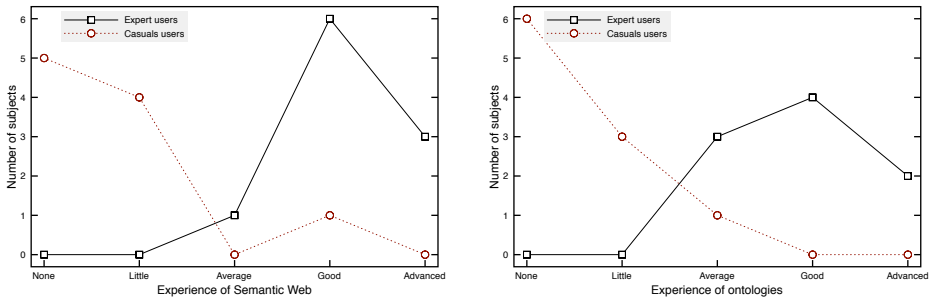
`dbo:deathCause` linking `dbo:Person` and `dbo:Disease` concepts. Finally, the most complex query *Show me all songs from Bruce Springsteen released between 1980 and 1990*, containing a date range constraint, was answered by no system, as reported by QALD organisers [CLU+13].

Note that the number of queries (five) was chosen based on the requirements presented in Section 6.2.1.2 and the literature review discussed in Section 4.4.2.4, to be sufficient enough to ensure representativeness and reliability of the evaluation, while balancing this with other aspects such as tiredness and overwhelming the recruited subjects as well as the resources required for executing the evaluation. As an example, six tasks were used in TREC-6 Interactive Track.

9.6.2 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 rewarded for their time. The casual users were drawn from the staff and student population of the University of Sheffield after the usability study was promoted on its relevant mailing lists. On the other hand, the expert users were drawn from the Organisations, Information and Knowledge (OAK) Group[17] within the Department of Computer Science at the University of Sheffield and from K-Now[18] – a software development firm, working on semantic technologies. The latter are all experts; having a fair amount of knowledge and experience in the Semantic Web field. Figure 9.9 shows a clear distinction between the two types of users in their knowledge of the Semantic Web and ontologies.

Note that some of the expert subjects recruited here have also participated in one of my previous studies. However, this could not have an affect on the results since this is a new query approach and system and thus they were new to it, like the rest of the subjects.



(a) Semantic Web

(b) Ontologies

Figure 9.9: User experience of the Semantic Web and ontologies.

At the beginning, subjects were introduced to the experiment and its goal, what is expected from them as well as any instructions required to be able to complete the

[17] http://oak.dcs.shef.ac.uk/
[18] http://www.k-now.co.uk/k-now/

experiment. Then, 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 then proceeded to the actual experiment in which they 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 (see Appendix C for details of all three questionnaires). Each full experiment with one subject took between 30 to 40 minutes.

Similar to the evaluations presented in the previous chapters, both objective and subjective data were collected covering the experiment results. 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, the *success rate*, capturing the percentage of tasks successfully completed, was used to measure effectiveness. Finally, subjective data collected through two post-search questionnaires was used to measure usability of the hybrid approach and satisfaction of users.

Finally, subjective data collected through two post-search questionnaires was used to assess the usability of the hybrid query approach and users' perceived satisfaction. The first is the *System Usability Scale (SUS) questionnaire*, used to investigate usability, while the second is the *Extended Questionnaire* which included a further question focusing on the ease of use of the hybrid approach in addition to two open-ended questions to gather additional qualitative data and 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?

9.6.3 Results and Discussion

To quantitatively analyse the data collected, SPSS[19] was used to produce averages, perform correlation analysis and check the statistical significance. The median (as opposed to the mean) was used throughout the analysis to calculate averages, since it was found to be less susceptible to outliers or extreme values sometimes found in the data. In the qualitative analysis, the open coding technique was used in which the feedback data was categorised and labelled according to several aspects dominated by usability of the query approach and implementation improvements suggested by the users.

[19] www.ibm.com/software/uk/analytics/spss/

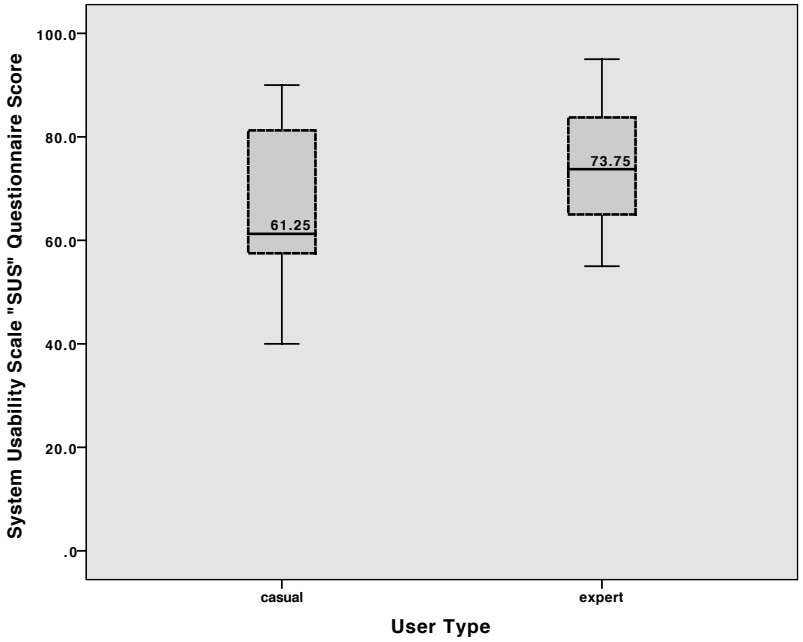


Figure 9.10: Average SUS scores for expert and casual users

According to the adjective ratings introduced by [BKM09] and the SUS scores shown in Figure 9.10, NL-Graphs is classified as *Excellent* by expert users (median: '73.75') and *Good* by casual users (median: '61.25'). Indeed, it would have been of great benefit to conduct the same evaluation – using the same dataset and questions – with a graph-based approach. This would have allowed comparing the results of both approaches resulting in a better basis for assessing the support provided by the hybrid approach for users during constructing their queries and whether this resulted in an improvement in the effort required in this process. However, unfortunately, my analysis and experimentations showed that, due to the complexity and structure of DBpedia (the evaluation dataset), formulating the evaluation queries using a graph-based approach would be very difficult – if not impossible – for users who are not domain experts with knowledge of the data and its structure. Additionally, in my view, a comparison with a NL-based approach is not suitable and would be biased since the queries were chosen based on the fact that systems employing such approach face problems in addressing (e.g.: difficulty in mapping query terms to ontological ones or highly complex queries).

Although the dataset used in the current evaluation (DBpedia) is different from that used in the usability study presented in Chapter 7 (Mooney) as well as the questions, I believe one could compare the SUS scores of NL-Graphs – employing a hybrid approach – achieved in the former to these of Affective-Graphs – employing a graph-based approach – achieved in the latter since the SUS questionnaire is mainly assessing users' satisfaction with the approach itself rather than their performance or timings in querying specific data or answering specific questions. Additionally, it would provide the reader with a more complete picture and contribute to the discussion of the results. Table 9.3 presents this comparison: both types of users had a more satisfying experience with

Table 9.3: Average (median) SUS score for NL-Graphs – from the current evaluation – and for Affective Graphs – from the usability study presented in Chapter 7.

Tool	**Expert Users SUS score**	**Casual Users SUS score**
NL-Graphs	73.75	61.25
Affective Graphs	63.75	55

NL-Graphs than with Affective Graphs, despite the domain being much more complex in the evaluation of the former.

These encouraging results are also supported by the success rate, informing effectiveness, and reported as 100%, showing that 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 in which only one casual user directly stated that she found the approach to be "*complicated and not intuitive*", while the other three users 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 this figure 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 previously shown in Section 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 to your questions, also, it is more interactive and shows options that you might not have considered exploring before.*
- *I find this mechanism to be highly useful for research in all areas of interest.*

Indeed, in my view, the casual users' experience and 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 Section 8, a system that is initially hard to learn could be eventually efficient [Nie93, p. 41]. This was also confirmed from both casual and expert users' feedback, shown below:

- *Once I got used to it, it was quite simple to use but if I was to start using it all the time I would like to have more training.*
- *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 think I was unfamiliar with the system and 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, some of the most repeated comments are as follows:

- *I thought the NL part was very straightforward to use and made a good starting point for constructing queries while the visualisations made it easy to realise the connections between the data.*
- *Providing the NL first was very quick and user friendly. This made it fast to formulate queries.*
- *It was useful to have all the relevant entities and classes preloaded onto the diagram.*
- *I liked that the system automatically identified the main concepts from the query so the exploration process was faster.*

Similar to both studies presented earlier, here I report the results of specific questions found in the SUS questionnaire which are focused on the usability and learnability aspects, as follows;

- I thought the system was easy to use.
- I found the system very tedious / troublesome to use.
- I would imagine that most people would learn to use this system very quickly.

Table 9.4: Scores given by users for individual SUS questions for NL-Graphs – from the current evaluation – and for Affective Graphs – from the usability study presented in Chapter 7. These questions are answered on a 5-point Likert scale ranging from *Strongly Disagree(1)* to *Strongly Agree(5)*.

Question (Strongly Disagree - Strongly Agree)	**NL-G: Expert**	**AG: Expert**	**NL-G: Casual**	**AG: Casual**
System easy to use	3.9	3.6	3.25	2.9
System tedious to use	1.7	2.1	1.9	2.3
Learn to use the system quickly	3.8	3.4	3.1	2.9

The first observation from Table 9.4 is that expert users (NL-G: Expert) were more satisfied with the usability and learnability of NL-Graphs than casual users (NL-G: Casual), which is inline with the previously discussed results and feedback. The other interesting piece of finding reported in the table is the comparison between NL-Graphs and Affective Graphs. As stated above, the overall SUS score showed that both types of users had a more satisfying experience with the first: (expert users: '73.75' compared to '63.75' and casual users: '61.25' compared to '55'). This is similarly shown here by the scores of the individual questions focusing on usability and learnability of the two systems.

Another output to report from this evaluation is with regards to the efficiency of the hybrid approach, assessed using the effort-based measures: *input time* and *number of attempts* required by users to formulate a query. 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. Again, as noted earlier, a direct comparison with Affective Graphs (employing a graph-based approach) is not possible due to the difference in this evaluation's dataset (DBpedia) from the one used in the mentioned usability study (Mooney). Yet, 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, expert users needed '88.86' seconds and '1.7' attempts while casual users needed '72.8' seconds and '1.5' attempts on average to construct a query. This view is also supported by our observations from both experiments as well as from users' feedback: the graph-based approach was judged as laborious and time consuming in the usability study presented in Chapter 7, while in the current evaluation of the hybrid approach, only three users commented on the effort required to build queries, which they found to be greater than in comparison with text-based search engines. Note that this is despite the domain being much more complex in the evaluation of the hybrid approach. Furthermore, most of the other users – especially experts – appreciated the hybrid approach for supporting them in building queries in a fast and straightforward manner (through the integration of the NL-component).

Figure 9.11 shows the average time required by users to formulate each of the five

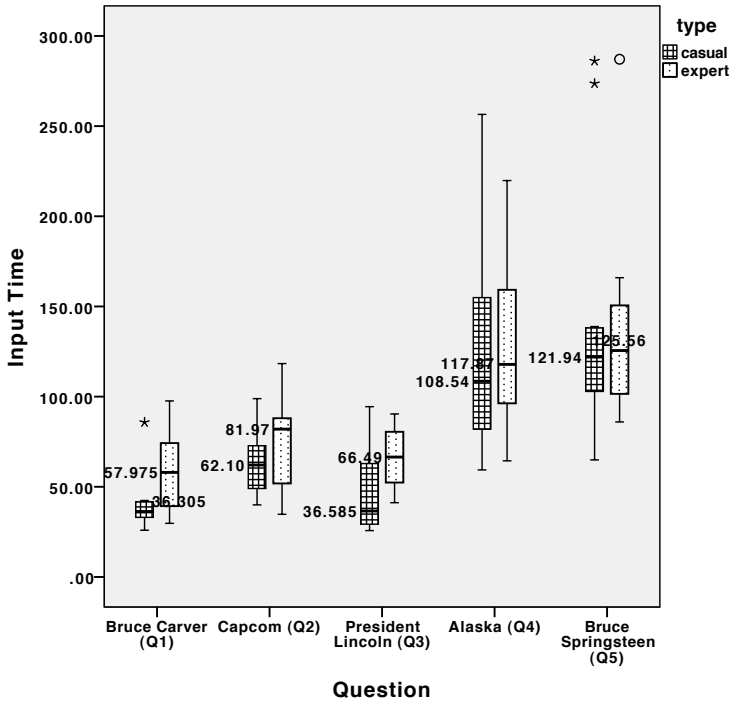


Figure 9.11: Average time required to formulate each question

evaluation queries. Firstly, this figure shows that the average time for all questions is negatively affected by the time required to answer the last two queries: *Alaska* and *Bruce Springsteen*. To understand the cause for the increase in the amount of time required, we used our observations and the screen recordings collected during the experiment and found the following:

- Give me all cities in Alaska with more than 10000 inhabitants: Firstly, few subjects attempted to use the query term *alaskan*, which was not recognised by Alchemy API and similarly by the NL-component, resulting in these users trying to set a constraint to the concept itself, a step which required an additional amount of time. Secondly, the DBpedia property `dbo:isPartOf`, connecting *Alaska* and the *cities* found in it, was confusing and not self-explanatory for users – even expert users – and they needed more time to check and think of 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 visual query and therefore users needed to add the constraint '*more than 10000*' to the property `populationTotal` using the visual approach. To accomplish this, three additional steps were required as shown in Figure 9.12.

  Again, this resulted in more input time for this query. As noticed, the second issue (concerned with `isPartOf` property) is related to the naming techniques used in the Semantic Web, while the third issue is regarding implementation details, which can be easily changed and therefore, I believe both issues are not considered as

Figure 9.12: Steps required to add the numerical constraint found in the query '*Give me all cities in Alaska with more than 10000 inhabitants*'. In the first step (1), the user right clicks the property and adds it to the query (Add/Remove Query), then in the second step, the user right clicks the property again to add a constraint (Add constraint), and finally, in the third step, the user adds the specific value for the constraint to the property as shown.

problems or effects of using the hybrid approach.

- 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`. As shown in Figure 9.13, this requires four steps. In each of the last two steps, the user has to use a date picker to specify the date required. 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, '1980 <date <1990', which was not available. Again, this issue is with regards to implementation details which can be improved and should not affect the usability of the hybrid approach.

The above scenarios show that adding numeric constraints found in queries is not yet automated – not added by the NL-component to the visual query which is automatically built. Indeed, the intention is to make it as automated as possible at a later stage with a more mature system. On the other hand, other constraints such as values of instances (e.g. Brooklyn Bridge) are directly added to the visual query. Additionally, during the experiment, it was observed that both casual and expert users happily and successfully formulated these queries containing the numeric constraints by adding the latter manually, despite taking higher input time that the other queries.

Secondly, Figure 9.11 shows that, on average, expert users took more time to build their queries than casual users. Again, observations and screen recordings showed two reasons that could explain this behaviour: 1) expert users followed logic and their understanding of the Semantic Web concepts to plan, formulate and validate their queries, which resulted in higher query input time; and 2) some expert users took more time to validate their queries using the formal (SPARQL) query presented in the interface and moreover, some of them used it to perform direct changes to their queries.

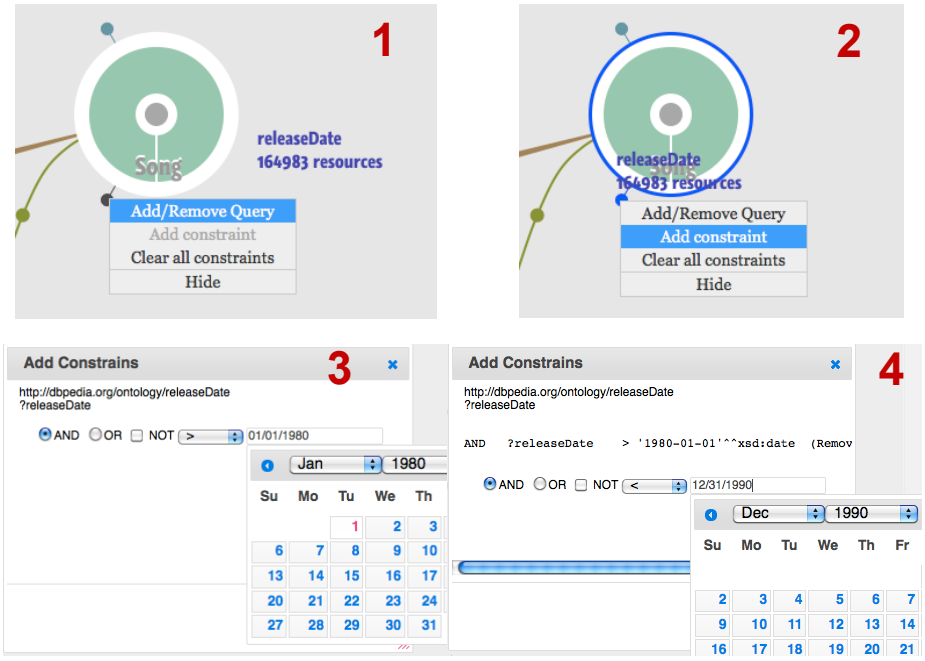


Figure 9.13: Steps required to add the date range constraint found in the query '*Show me all songs from Bruce Springsteen released between 1980 and 1990*'. In the first step (1), the user right clicks the property and adds it to the query (Add/Remove Query), then in the second step, the user right clicks the property again to add a constraint (Add constraint), and finally, in the last two steps, the user adds the specific values for the date range constraint as shown.

Query Validation

As illustrated in Section 9.3.2, 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 the query validation and correction feature proved to be very useful and helpful for users while constructing their queries. Indeed, the screen recordings showed 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 (see Figure 9.14). 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 properties and concepts identified so that I didn't need to click on them a lot in the visual interface to do changes.*

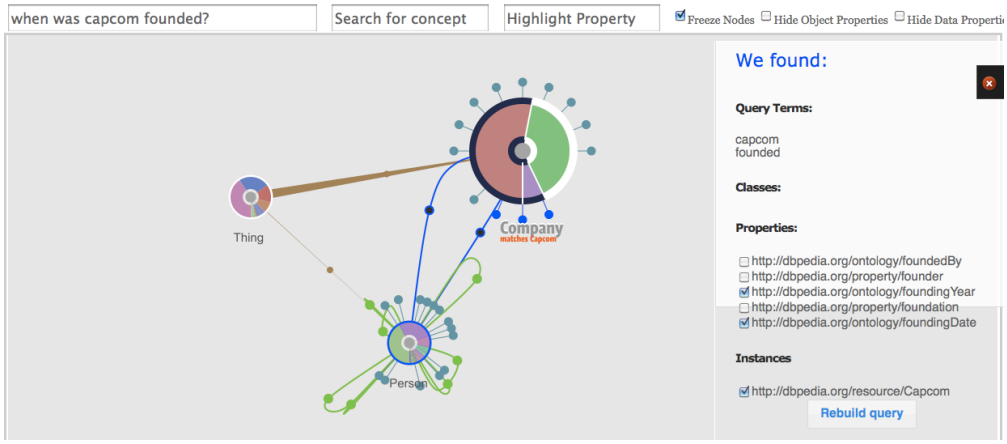


Figure 9.14: Validation and correction of the input interpretation of NL-Graphs for the query "*when was capcom founded?*".

- *The options to validate and refine searches were obvious and well set out.*

9.7 Summary

In this chapter, I have presented the hybrid query approach which was motivated by the outcomes of the studies presented in Chapters 7 and 8. The approach 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 prototype of the approach, called NL-Graphs, was presented which comprised *Affective Graphs* – the most liked view-based approach in the usability study – and a NL-component specifically developed for this purpose. The architecture of the approach was explained, together with illustrative scenarios showing the querying experience in NL-Graphs. Additionally, the usability study conducted to assess the usability of the approach and its usefulness in supporting users during query formulation was presented. In this study, 24 subjects (12 expert users and 12 casual users) were asked to perform five search tasks. DBpedia dataset was used together with a set of queries with which NL-based approaches would face problems while attempting to answer. The queries were selected from the 2nd Open Challenge in QALD's workshop. To assess the usability, efficiency and effectiveness of the approach and users' satisfaction, both objective data – such as the input time, number of attempts required to answer a question and the success rate – and subjective data – users' input for post-search questionnaires, observations and screen recordings – were collected and quantitatively and qualitatively analysed.

The results of the evaluation are very encouraging, with both types of users providing 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 questions given in the study. Additionally, answers to the question regarding the *query construction process* in the extended questionnaire showed that most users could *effortlessly* use the hybrid approach (as a query mechanism) to formulate and answer questions. Indeed, this was

also observed from the collected feedback: 19 of the positive comments were focused on the usability and support provided by the hybrid approach during query construction; and only four users provided negative feedback, three of which were due to the longer time or more steps required to build queries than with text-based search engines such as Google. I believe these encouraging results provide a good basis and motivation for future work towards deeper investigation into hybridising semantic search systems and their resulting performance.

Finally, I believe that NL-Graphs is only a step forward in this direction, yet, there is much room for improvement. Firstly, the design of the NL-component presented earlier is a templates-based approach for matching users' queries with a set of predefined templates. Indeed, this approach has been gaining more attention recently (as discussed in Chapter 3 and used for instance in TBSL) for its potential, however, relying on fixed templates on the other hand is not guaranteed to provide a suitable match for all types of questions. Additionally, queries with numeric constraints is not yet automatically added by the NL-component to the visual query, but requires user engagement. Finally, the NL-component is affected by issues related with noise and quality of the data, such as unspecified property universes, similar to other SOA approaches.

However, the advantage in NL-Graphs (adopting the hybrid approach) is that, in such scenarios, where it is difficult for the NL-component to find matches for specific terms, the user will still be able to use the visual approach to proceed in formulating queries. Unfortunately, for other issues such as untyped entities, currently there is no specific solution for this problem, which indeed is a challenge for all different SW applications consuming this data.

Part III

Conclusions

"*Success is not final, failure is not fatal: it is the courage to continue that counts.*"

– Winston Churchill

Chapter 10

Conclusion

10.1 Summary of Findings

Usability and user satisfaction are of paramount importance when designing interactive software (including semantic search) solutions. Furthermore, the optimal design can be dependent not only on the task but also on the type of user. Despite this, evaluating semantic search tools with respect to these aspects in order to develop more user-oriented approaches has been fairly overlooked. Until we understand and further investigate this, improving search mechanisms (to provide better performance) would not be sufficient enough to unleash the full potential of semantic search for end users. This was the motivation behind the work presented in this thesis. Therefore, the main research question was the following: "*How can we design a user-oriented semantic search query approach that is effective and usable beyond current state of the art approaches?*". Different pieces of work were then conducted to explore more specific questions whose answers together facilitated investigating the main research question.

In order to answer the above research question, it is first important to understand users' needs and preferences and how to cater to them. Therefore, the study presented in Chapter 7 served this purpose and explored the question: "*How do casual and expert users perceive the usability of different semantic search query approaches?*". To answer this question, five semantic search tools employing four query approaches (free-NL, controlled-NL, graph-based and form-based) were evaluated. Twenty subjects (10 expert users and 10 casual users) participated in the experiment which followed a within-subjects design to allow direct comparison between the evaluated query approaches. Each subject was asked to perform five search tasks querying Mooney geography dataset. The simplicity of the domain was the main criterion for selecting this dataset, for users to be able to understand and formulate the given questions into the tools' query languages. The questions included simple as well as complex ones such as "*Give me all the capitals of the USA?*" and "*Which states have a city named Columbia with a city population over 50,000?*" respectively. To assess the usability of the approaches and users' satisfaction, objective data – such as the input time and number of attempts required to answer a question – and subjective data – users' input for post-search questionnaires

– were collected and quantitatively and qualitatively analysed. The results of this study revealed the strengths of view-based approaches in supporting users during query formulation. Indeed, visualising the search space helped users to understand the data and the possible ways of constructing queries. However, unsurprisingly, the main drawback of these approaches was found to be the high query input time and user effort required during query formulation. The study also showed that the flexibility, ease of use and expressiveness offered by free-NL was highly appreciated.

The outcomes of the previous study showed that despite the highest satisfaction of users by view-based approaches, they were found to require a fair amount of effort and time in constructing queries, which could affect their usefulness while performing the intended search tasks. However, since the use of some systems employing these approaches is expected to require an amount of learning, assessing learnability was deemed essential. Therefore, the user-based study presented in Chapter 8 attempted to investigate the learnability of the best-performing view-based system. The work thus explored the following research question: "*Can training and frequency of use of a query approach improve the proficiency level and efficiency of users (in terms of time and effort) in answering search tasks of different complexity?*".

To answer this question, *Affective Graphs* – the most liked tool adopting a view-based approach in the usability study presented earlier – was selected for this study which was conducted with ten expert users over three different evaluation sessions. The users were asked to perform 12 search tasks (four different tasks in each session) in these sessions which took place over three consecutive days. The Semantic Web Dog Food (SWDF) dataset in addition to real world queries were used in this study. Again, simple as well as complex queries were used, such as "*Give me the people with first name 'Knud'*" and "*Give me the name, homepage and page of people who were workshop organisers for a workshop about 'Ontology Matching'*" respectively. In order to assess the users' performance, objective data such as query input time and number of attempts required for answering each task was recorded. Additionally, users' search behaviour and strategies were observed throughout the evaluation sessions and finally, their experience was captured using questionnaires. The collected data was quantitatively and qualitatively analysed to assess learnability and satisfaction. The results of the study showed an improvement in users' performance, reflected in a decrease in the query input time (Session 1: '106.3' and Session 3: '61.6' seconds). Furthermore, the results showed an increase in users' satisfaction reflected in the average SUS score which increased from '76.25' to '82.5'. In spite of these positive outcomes, the effort and input time required (even after the improvement) during query formulation could still be an issue for users with frequent search tasks.

The outcomes of the studies presented above motivated the design of a hybrid query approach that takes advantage of visualising the search space offered by view-based query approaches and the ease of use and speed of query formulation offered by free-NL. NL-Graphs, the implementation of this hybrid query approach, presented in Chapter 9 combined *Affective Graphs*, the most liked view-based approach, with a NL-component specifically developed for this purpose. To test my hypothesis for the usability and

usefulness of the hybrid approach in supporting users in finding answers for their information needs, NL-Graphs was evaluated in vivo with both casual and expert users. Twenty-four subjects (12 expert users and 12 casual users) participated in the experiment and were asked to perform five search tasks. One of the main differences from the studies discussed earlier is the choice of DBpedia dataset. This choice was intended to increase realism of the study – DBpedia is more representative of real data found on the Semantic Web in terms of data quality, heterogeneity and noise – and indeed, to evaluate the usability of the hybrid approach and users' experience whilst querying a large complex dataset. Another difference is with respect to the questions used, since the aim was to select a set of queries with which NL-based approaches would face problems while attempting to answer. This would help in assessing the usefulness and worth of the hybrid approach in supporting users in finding answers for such queries. Therefore, five queries were selected from the 2nd Open Challenge in QALD's workshop. To assess the usability, efficiency and effectiveness of the approach and users' satisfaction, both objective data – such as the input time, number of attempts required to answer a question and the success rate – and subjective data – users' input for post-search questionnaires – were collected and quantitatively and qualitatively analysed.

The results of the evaluation are very encouraging, both types of users provided high SUS scores for NL-Graphs – with experts being more satisfied. Success rates also showed that all users were able to successfully answer all the questions given in the study. Additionally, answers to the question regarding the *query construction process* in the extended questionnaire showed that most users could *effortlessly* use the hybrid approach (as a query mechanism) to formulate and answer questions. Indeed, this was also observed from the collected feedback: 19 of the positive comments were focused on the usability and support provided by the hybrid approach during query construction; and only four users provided negative feedback, three of which were due to the longer time or more steps required to build queries than with text-based search engines such as Google. The latter is an expected outcome and should not be discouraging since the rest of the users' feedback was indeed positive:

- Casual users appreciated the visual approach (perceiving it as the added-value to the text-based approaches) and found it highly useful and helpful in showing the data and its structure (relationships between concepts), understanding what they can search for, providing context for the query and options to explore and finally creating an interactive and interesting search experience.

- Expert users had more appreciation for the support provided by the NL-component (perceiving it as the added-value to the graph-based approaches) and found it very quick, user-friendly and straightforward to use. In addition, having its output (relevant entities, concepts and properties) automatically visualised led to a faster approach for constructing their queries and in turn to high user satisfaction.

These encouraging results of evaluating the hybrid approach – which was an outcome of most of the work done throughout the thesis – show that this work and the whole thesis provide a good basis and motivation for other researchers for a deeper investigation

into hybridising semantic search systems and their resulting performance. I hope this could result in more progress in the area of semantic search and in reaching a wider population of users, not limited to the Semantic Web community.

Whilst answering the research questions presented above, this thesis additionally made practical as well as scientific contributions. First of all, the usability study presented in Chapter 7 provided direct comparison of different query approaches and a first-time understanding and comparison of how expert and casual users perceive the usability of these approaches. The results and findings of this study contribute great value for the Semantic Web community, especially for developers of future query approaches and similar user interfaces who have to cater for different types of users with different preferences and needs. Secondly, the learnability study presented in Chapter 8 is the first work to investigate and address learnability of a view-based query approach and how it influences performance, proficiency and satisfaction. Both studies also highlighted the need within the semantic search community to move towards more comprehensive views of semantic search evaluations by addressing these criteria (usability and learnability) which are as important as the retrieval performance.

A third contribution of this thesis is the preliminary work (presented as part of my future work since it is not yet evaluated) to identify information needs of users querying the Semantic Web and Linked Data. This work, presented in Chapter 11, is a first-time analysis of semantic query logs to identify these needs. I believe the findings of such an analysis are beneficial for researchers and developers, especially linked data providers who would benefit from matching their data with the needs of linked data consumers. Additionally, the analysis provided insights into the patterns and trends inherent in users' queries. In my view, this reveals great potential, not only for semantic search, but also for different Semantic Web applications which could benefit from having an advance knowledge of the most queried categories and the associated search patterns followed by users. Fourth, an approach for using the previously mentioned semantic query logs to return more information for users with the results has also been proposed in the same chapter. The approach was motivated by the findings of the usability study which showed the users' need for more information returned with the search results to provide a richer experience and a wider understanding. The strength of the proposed idea lies in utilising query logs as a source of collaborative knowledge, able to capture perceptions of Linked Data entities and properties, and use it to select which information to show the user rather than depending on a manually or, indeed, randomly predefined set.

Moreover, a theoretical contribution of this thesis is the comprehensive review of the literature of semantic search, which was presented in Chapter 3. Semantic search is still in its infancy and many of the current approaches are facing challenges that were discussed throughout this thesis. The review thus provides a fundamental contribution for understanding the strengths and weaknesses of the different approaches, which is necessary for further progress and improvements. Another contribution is the review of SOA in semantic search evaluations and the analysis provided in Chapter 5, which highlighted the most important limitations and missing aspects in these evaluations.

Based on this analysis and my experience, recommendations and best practices proposed to support the semantic search community in tackling these limitations and filling the identified gaps are discussed below. I believe the review together with the set of lessons and practices are beneficial in fostering research and development in the field.

10.2 Best Practices for Running Semantic Search Evaluations

10.2.1 Datasets

10.2.1.1 Size

The size of the dataset should be large enough to be representative of datasets currently found on the Web of Data. This trend can be observed in the IR community (e.g. currently, TREC uses corpora of up to a billion documents[1]) and the growing emphasis on 'Big Data' in general means insightful evaluations must incorporate such datasets. Examples of *single/closed-domain* datasets (ones which describe a single domain) currently found on the Web of Data are *Geonames*, *PubMed* and *LinkedGeoData* which contain around 150 million, 800 million and 3 billion triples, respectively. Therefore, for a single-domain evaluation scenario, a dataset of less than 100 million triples would be small, between 100 million and 1 billion triples is acceptable and more than 1 billion triples can be required in some cases. Additionally, examples of *multiple/open-domain* datasets (heterogeneous ones spanning various domains) are DBpedia and Sindice 2011 which contain 1 billion and 11 billion triples, respectively. Therefore, for an open-domain evaluation scenario, a dataset of less than 1 billion triples would be small, between 1 billion and 10 billion triples is acceptable and more than 10 billion can be required in some cases.

10.2.1.2 Origin

[SJVR76] suggested that an ideal test collection should contain documents that vary in their source and origin; we believe a dataset for semantic search evaluations should also contain data collected from different sources, including triples from the datasets in the LOD cloud as well as semantic data gathered from different domains on the Web of Data.

10.2.1.3 Quality

Datasets found on the Web of Data are known to contain a certain level of noise and erroneous data, especially the operationally-derived datasets such as DBpedia. In contrast, specially-created datasets such as Mooney (described in Section 4.5.1) are usually of higher quality. Ideally, evaluations would use datasets featuring different levels of quality for assessing semantic search approaches in different scenarios. For instance, operationally derived datasets can be used to test their ability to work with data found

[1] http://plg.uwaterloo.ca/~trecweb/2012.html

in the real-world while specially-created datasets can be used when they ought to have specific characteristics, for example, to simulate high-quality datasets found in some organisations.

10.2.1.4 Data Age

Similar to how the size of the datasets used in evaluations should be representative of datasets found in the real-world, these datasets should also be up-to-date to capture any changes in the Semantic Web standards and technologies. Outdated datasets can directly affect the reliability of evaluations.

10.2.2 Queries

10.2.2.1 Size

The number of queries used should be large enough to produce reliable results especially since the stability of evaluation measures is directly influenced by this number. It has been common to use 50 queries in IR evaluations[2] and similarly in the semantic search evaluations reviewed in this thesis (see Table 4.1).

Therefore, for system-based evaluations (focusing on assessing retrieval performance) which can be done in an offline mode, between 50 and 100 queries would be acceptable. In contrast, in user-based evaluations, this number is usually much smaller since it directly influences the amount of resources (time, cost, effort) required and the subjects recruited. Most of the IIR and semantic search user-based studies discussed above used between four and 20 queries. However, using a large number of queries such as 20 was found to be too many for the subjects [WER+10]. Wrigley et al. explained that in the final quarter of the experiment, the subjects tended to become tired, bored and sometimes frustrated. Therefore, we believe a best practice would be to use a number of queries in the range of 5 and 15. If more queries are necessary for a specific scenario, multiple sessions could be used to alleviate the previously mentioned effects on subjects.

10.2.2.2 Origin

Ideally, the queries used should be real, describing genuine user information needs. These ought to be collected from logs of different state-of-the-art semantic search systems in order to provide a breadth of query formats, information targets, etc. However, given the infancy of the field, this is challenging – there currently aren't enough human-focussed systems or users to provide representative query histories (let alone users who are 'non-experts' in the semantic web field). An alternative source are the semantic search engines (such as Sindice [TOD07]) and federated query architectures (such as SQUIN [HBF09]) which are commonly used programmatically by other Semantic Web applications. The problem with using their query logs is twofold. First, the tasks/use cases directly influence the characteristics of the queries (see Section 5.2) and are usually very different from queries issued by human users. The second problem is with respect

[2] http://sites.google.com/site/trecfedweb/
http://plg.uwaterloo.ca/~trecweb/

to the representation of the queries. Most of the queries issued to the first type of applications (semantic search engines) are usually given as keywords, while those issued to the second type of applications (federated query architectures) are usually written in SPARQL. On one hand, keywords can lack important information such as the relations between concepts or entities found in the queries which can affect the subjects' understanding and interpretation. On the other hand, SPARQL queries are not suitable for subjects who are not Semantic Web experts. Therefore, there is a step required to translate either of these types of queries into NL verbalised statements to be used in semantic search evaluations.

10.2.2.3 Complexity/Difficulty

Since evaluations aim to assess systems in real-world scenarios, a requirement is, therefore, to use queries of different levels of complexity (e.g. different number of concepts and properties) and comprising different features (such as negation) that are typically found in real queries. Using only simple or complex queries could affect the realism and in turn reliability and efficacy of the evaluations.

10.2.2.4 Type

Queries can be broadly categorised into *factual queries*, in which the information needed is about a particular fact, and *exploratory queries* in which the information need is more generic and does not specify a particular fact. According to the classification used in Section 4.4.2.4, the first type covers both *specific fact-finding* and *extended fact-finding* tasks, while the second type covers *open-ended browsing* and *exploration* tasks. An example of a factual query is "*Give me all the capitals of the USA*" (taken from Mooney), while an example of an exploratory query is "*Provide information on the Bengal cat breed*" (taken from TREC). While both types have been used in IR evaluations, semantic search evaluations have been mostly adopting factual queries. A justification for using this type of query could be related to the current ability of semantic search approaches in coping with exploratory queries. A more probable justification is the fact that it is much easier to generate groundtruth for factual queries which allows measuring precision and recall for the evaluated approaches. Indeed, it is very challenging to generate groundtruth for exploratory queries since it is not clear what would represent a *correct* answer. In this scenario, the same approach adopted in IR can be used in which human judges are asked to evaluate a sample pooled from the top results of different retrieval systems to construct the final groundtruth. Yet again, the difficulty here would be to determine the relevance of an entity URI or a literal value, which are the types of answers returned by semantic search systems, to the given query, as noted by [PMZ10]. This decision can be highly subjective which would increase the number of judges required to allow a level of inter-judge agreement. Altogether, this causes the evaluation process to be resource intensive (in terms of time, effort and money). We believe this challenge should be addressed since both types of queries are found in the real-world search scenarios and represent genuine information needs.

10.2.2.5 Representation

Although queries were limited to verbal statements in some IR studies in literature [Cle70], a common approach currently used in most IR evaluations (such as TREC) is to include this verbal statement together with other information such as a description and a narrative in the so-called *topic*. Semantic search evaluations have been using only verbal statements to describe their queries. Topics provide more information which helps subjects/judges identify the relevant answers in the results. Additionally, as discussed earlier, the *simulated work task situation* which was proposed by [BI97] provides more realism by describing the situation that led to the information need. Indeed, the semantic search community should investigate the possibility of using these representations to increase the reliability of evaluations and their results.

10.2.3 Groundtruth

Some of the semantic search evaluations discussed above generate a SPARQL query equivalent to the NL query and execute the former to produce the groundtruth. However, this might not produce very accurate results since the mapping from the NL to a SPARQL query is manually performed by the organisers and is subjective; there is not always one *right* way to do it since there can be different paths in the dataset that lead to the same result (concepts can be linked to each other through different paths). It is therefore difficult to guarantee the completeness of the groundtruth generated by following this approach. This could result in some of the evaluated systems to have recall *less than one* if they follow different paths, generate different SPARQL queries and therefore get different results, which may still be relevant to the query. We believe that a more accurate approach could be to inherit the pooling technique from IR, in which different systems are asked to submit their results and the top K results are merged and assessed by human judges to generate the groundtruth. Recently, crowdsourcing this and similar tasks has received an interest within the research community, for example, using Amazon Mechanical Turk. However, this should be further investigated since the feasibility and reliability of this approach are not yet agreed on [Kaz11, CLY11, CST+12].

10.2.4 Evaluation Criteria and Measures

Ranking is a necessity for search. It is important to encourage adopting ranked-based measures (as opposed to set-based measures such as precision and recall). Also, it is important to distinguish between systems based on their different levels of performance in retrieving relevant answers. Graded-relevance scale and the suitable measures (e.g. nDCG) should be used (as opposed to 'relevant' and 'non-relevant' mode). Indeed, this is more difficult to achieve with semantic search tools returning specific answers for factual queries (e.g. 'capital of a given state'). However, as earlier mentioned, these guidelines and best practices are intended to support both large-scale semantic search initiatives and smaller-scale exercises such as individuals or companies interested in evaluating their own systems, and therefore have a general scope, rather than targeting a specific tool type.

Moreover, relevance assessment should be performed by human judges, with careful consideration to what affects judges' assessments, especially key aspects such as the difference in their knowledge (experts in the domain versus non-experts) or the order of presentation of the results (which can be normalised by randomising this order). Also, measures ought to be chosen while taking into account the number of queries used in the evaluation design and the number of results that will be assessed – per query – since both aspects influence the stability of the measures used.

10.2.5 User-based evaluation

In designing user-based evaluations of semantic search systems, the following aspects are important and require careful consideration.

10.2.5.1 Evaluation Setup

Running user-based studies is known to have a high overhead with the need to allocate resources (time, cost, effort) and recruit subjects. This is in addition to the logistics required for carefully organising and scheduling an evaluation, as well as the overhead incurred in the data analysis which is acknowledged to be extremely time consuming and labor-intensive [Kel09]. This process is, indeed, more difficult when evaluating more than one system. This has led to researchers refraining from evaluating their own systems in a user-oriented scenario, let alone comparing them with others. Thus, having a central organisation responsible for evaluating different systems and approaches is highly required. It would also facilitate adopting a within-subjects design to allow comparison of results. Finally, it has the advantage of guaranteed fairness of the evaluation process since systems would be explained in equal time periods and by external people, which sidesteps any possible bias that could be introduced by having developers evaluating their own systems.

In addition to the requirements specified above for the choice of the evaluation dataset, my experience with user-based evaluations raises another issue to consider. I found that inconsistencies in the dataset as well as naming techniques used within the Semantic Web community could affect the users' experience and their ability to perform the search tasks and in turn the evaluation's results and reliability. For instance, users in one evaluation were confused with inverse properties (e.g. *runsthrough* and *hasRiver* found in Mooney dataset) when they were shown to them while building queries using a view-based query approach. Similarly, a property like `dbo:isPartOf` (found in DBpedia), connecting places like regions and cities found in them, was confusing and not self-explanatory for users. This introduces an additional difficulty while choosing a dataset to ensure a balance between evaluation realism (by choosing datasets representative of those found in the Semantic Web) and evaluation reliability (by trying to avoid these characteristics in the chosen datasets).

Moreover, in the choice of the evaluations subjects, it is mostly important that they suitably represent the population, which mainly depends on who is targeted by the evaluated systems. In literature, users have been usually categorised as expert users and

casual users (see Section 4.4.2.3). Lots of systems developed within the Semantic Web community have been evaluated with its experts. This is usually due to the difficulty of finding casual users who are able to understand and assess these systems. However, this needs careful consideration since ideally, the goal for the Semantic Web and similarly semantic search is to reach a wider population of users, not limited to the Semantic Web community. Indeed, evaluating systems with both types of users and comparing their results is beneficial and could provide an understanding of the suitability of certain search approaches to specific types of users and furthermore, the different requirements and preferences to cater for when targeting a particular type of users. Additionally, deciding the number of subjects to recruit for a user-based evaluation is an open question and is influenced by the availability of resources (time, cost, effort) and subjects with the required characteristics. It can also affect the reliability of the evaluation results. Based on IIR and HCI literature (see Section 4.4.2.3) and also our experience, we believe that a number ranging between 8 and 12 subjects would be acceptable.

Finally, with respect to data collection, in addition to using individual questionnaires to assess certain aspects of each system, we found that an overall questionnaire (presented after evaluating all the approaches) asking the user to rank the systems with respect to certain aspects can produce more accurate comparisons since the rankings are an inherently relative measure. Such comparisons using the individual questionnaires may be less reliable since the questionnaire is completed after evaluating each approach's experiment (and thus temporally spaced) with no direct frame of reference to any of the other approaches.

10.2.5.2 What to evaluate

As argued in IIR literature (see Section 4.4.1), utility could be a more appropriate measure for evaluating IIR systems and their ability to support users in their search tasks. Assessing utility and usefulness of results as opposed to relevance would capture how the user judgment is influenced by other aspects beside the relevance of the answer to the query. Examples of these aspects are users' background and knowledge (what is already known about the query subject); the interaction between the system and the user; or the representation of the answer itself (it can be understood for instance by one user and not by another). [Gof64] notes that "any measure of information must depend on what is already known". Therefore, to assess utility, one could use questions with an overall goal – as opposed to ones which are not part of any overarching information need – and compare users' knowledge before and after the search task. Since the usefulness of a result in this scenario will be evaluated by the user, exploratory queries could be used in addition to factual queries since there is no need to worry about generating the groundtruth for the queries. Furthermore, as discussed earlier, using *simulated work tasks* is intended to develop simulated information needs by allowing for user interpretations of the situation. All of the above together would, indeed, add more realism to the evaluation and increase its reliability.

Moreover, one of the mostly used evaluation criteria in IIR and usability studies is efficiency (commonly assessed using time- or effort-based measures). From previous

evaluations, we found that both measures should be used in order to obtain a full impression of efficiency (either measure alone provided only a partial account). For instance, the time required by users to formulate queries in a system can be low but the number of trials performed to answer a question is high. In such a situation, using both measures would provide more reliable results and comparisons. Additionally, we believe that it is important to evaluate system-level efficiency (e.g. response time) since this strongly influences user satisfaction. Despite its importance, this aspect has been omitted from previous user-based semantic search evaluations.

Furthermore, any new interface, application, or a product is expected to require some amount of learning. [Nie93] notes that a system that is initially hard to learn could be eventually efficient. Certainly, investigating the ease of learning how to use a system would be even more beneficial when evaluating a new or advanced approach or interface that users are not familiar with (such as the different visual approaches consuming Linked Data). This can be achieved by evaluating *extended learnability* which refers to the change in performance over time [GFA09] as opposed to *initial learnability* which refers to the initial performance with the system. This aspect, despite its importance, has been missing from user-based evaluations of semantic search systems. Studies focusing on extended learnability are usually referred to as *longitudinal* studies, which are conducted over an extended period of time, with evaluation measures taken at fixed intervals, both of which determine the number of sessions required [Kel09]. It is thus important to decide on this period of time as well as the interval between the sessions. Similar to the choice of the number of subjects required for a usability study, the number of sessions presents a tradeoff between reliability of the evaluation (since it directly affects the amount of collected data and results), and its overhead. On the other hand, the interval between two evaluation sessions should be influenced by the expected/actual use of the evaluated system or interface. For instance, since search tools are often used everyday, the evaluation sessions should be placed over consecutive days (with the same users).

10.2.6 Repeatability and Reliability

Repeatability and reproducibility of an evaluation is concerned with whether its repetition over a period of time produces the same results and rankings of systems. A main factor in achieving this repeatability is the control over the experimental variables. Hence, the user-oriented approach to evaluations has been acknowledged to face difficulties with being repeatable. One of the main reasons is the variability introduced by human factors such as differing search behaviour and strategies, their capabilities in expressing the information need, as well as their satisfaction levels and criteria. On the other hand, in the system-oriented approach the main factor is the consistency of relevance assignments for a specific result.

As discussed in Section 10.2.3, in the SEALS and QALD evaluations, generating groundtruth for a specific query is performed by running its equivalent SPARQL query over the evaluation dataset. Although we argued against the reliability of this approach,

it guarantees the repeatability of the results. Indeed, this requires using the exact query on the same version of the dataset to avoid any changes in the resulting assessments. In contrast, TREC and SemSearch used human judges to assess the relevance of results. The repeatability here thus depends on the degree of inter-judge agreement. The difference between the two evaluations is that TREC used expert judges whereas SemSearch used Amazon Mechanical Turk workers in the assessment process. [BHH+13] pointed out the limitation – in terms of scalability – of depending on a limited number of expert judges since, in repeating the evaluations done by other researchers, it would be difficult if not impossible to use the same judges. Additionally, they showed that repeatability was successfully achieved through crowdsourced judgments since conducting the same experiment with two different pools of workers over a six-month period produced the same assessments and same rankings for the evaluated systems.

Another important aspect that could influence repeatability is the cost of conducting an evaluation. [CST+12] note that crowdsourcing potentially lowers this cost, and thus, is an advantage of using this approach. They conducted an experiment in which they showed that the cost of recruiting 73 workers on Amazon Mechanical Turk, for around 45 hours to judge the relevance of 924 results, was $43.00, while the expert judge cost was $106.02 for around 3 hours of work. Additionally, Blanco et al. showed that, using Amazon Mechanical Turk, an entire SemSearch challenge was conducted for a total cost of $347.16. In this competition, 65 workers judged a total of 1737 assignments, covering 5786 submitted results from 14 different runs of 6 semantic search systems. Blanco et al. thus considered this approach to be cost-effective. However, arguably, being "*cost-effective*" is very subjective: while it could be affordable for an organisation or an evaluation campaign, it is more likely to cause difficulties for an individual researcher (e.g. a PhD student) and thus affect the repeatability criterion.

With regards to reliability, we illustrated how the approach adopted by SEALS and QALD is the most problematic since it does not guarantee the completeness of results and, in turn, the reliability of the assessments. The use of expert judges (as in TREC) is found on the other end of the spectrum as the most reliable approach. However, due to the issues with this approach described above (scalability limitation and high cost), several experiments have recently been conducted to investigate the reliability of using non-expert judges (e.g. Amazon Mechanical Turkers) as an alternative. However, a consensus on this approach's reliability does not yet seem to have been reached. On one hand, [AM09] showed that crowdsourcing was a reliable way of providing relevance assessments, the same conclusion of a more recent study by [CLY11]. On the other hand, [CST+12] and [BHH+13] showed that, while crowdsourced assessments and expert-judges' assessments produce similar rankings of evaluated systems, they do not produce the same assessment scores. Blanco et al. found that, in contrast to experts who are pessimistic in their scoring, non-expert judges accept more items as relevant. Additionally, the level of inter-judge agreement was much higher for expert judges than for non-experts (0.57 versus 0.36). Despite this, they concluded that the reliability of non-expert judges is still sufficient since they provided the same overall ranking of the systems as the expert judges. We suggest, however, that the ranking of systems is not

the sole goal of an evaluation: understanding and investigating the real performance of systems is equally important. Indeed, [CST+12] note that crowdsourced workers are not guaranteed to provide assessments that enable accurate measurement of the differences between two systems. Furthermore, being lenient in the assessment process and producing high results for evaluated systems could, indeed, mask the systems' true performance.

Finally, it is important to reemphasise the need for more work towards evaluation platforms that enable persistent storage of datasets and results that guarantee their reusability and the repeatability of tests. Forming agreement on the approaches and measures to be used within the search community for evaluating similar categories of tools is a key aspect of standardised evaluation. In addition to the resources created, the value of organised evaluation campaigns in bringing together members of the research community to tackle problems collectively would help in accelerating progress in the semantic search field.

Chapter 11

Future Work

The motivation of the work presented in this thesis was to improve the usability of semantic search query approaches from the perspective of end users. Therefore, the usability and learnability studies presented in Chapters 7 and 8 attempted to provide an insight into the requirements and needs of these end users and their levels of satisfaction with the different query approaches. The work described in these chapters raised some ideas for future work which are outlined in the rest of this chapter. Some of these ideas have been analysed and implemented but not yet evaluated and others are only proposals by the author.

11.1 Exploring Users' Information Needs from Query Logs

In addition to understanding users' preferences and needs and their satisfaction with different query approaches – which was done in the studies presented earlier – it is also important to identify what information they are looking for and interested in. Therefore, this section (Section 11.1) presents work done which serves this purpose[1]. It is based on analysing query logs to explore this research question.

During the last two decades, traditional search engines have improved in precision and recall by aiming at matching the Web's content with the information needs of Web users. Part of this progress has been possible thanks to the analysis and interpretation of query logs [SMHM99, JS05]. These studies addressed statistics involving query length, term analysis and topical query classification [HS00, Spi02], as well as the identification of changes in users' search behaviour over time [JSP05]. However, the nature of traditional query logs limits the analysis to a set of timestamped keywords and URIs, which lacks structure and semantic context.

The movement from the Web of documents towards structured data has made significant progress in recent years. Semantic Web gateways such as Sindice [TOD07] and

[1]This work was done in collaboration with Elizabeth Cano and Suvodeep Mazumdar. As part of my future work, I have worked with them to do this only in a preliminary format, and based on our discussions, they have implemented the visualisation toolkit (SEMLEX) explained below

Watson [dBG+07]) expose SPARQL endpoints which allow performance of more complex queries and reasoning over the Web of Data. Although the use of these gateways has built up a rich semantic trail of users' information needs in the form of semantic query logs, little research has been done on the interpretation of query logs as clues for analysing and predicting information needs at the semantic level. Existing studies have focused on metadata statistics derived from Semantic Web search engines [MMZ09].

This work explores questions such as: 1) what information do individuals or software agents look for on Linked Data? and 2) how do they query Linked Data to answer their information needs? Such analyses can give an insight into the coverage and distribution of queries over the data and whether people are making use of the whole or just a small portion of a dataset. Additionally, it can support in identifying interesting trends or hidden patterns in users' queries. To facilitate the discussion, we define information needs – in this work – as the set of concepts and properties users refer to while using SPARQL queries. Instances are dereferenced by querying the linked data endpoint for the type of the instance to identify which concepts the user's query is focused on. To illustrate, consider the following example:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?manufacturer
WHERE {
     <http://dbpedia.org/resource/Acura_ZDX> dbo:manufacturer ?manufacturer.
}
```

The example query shows a user looking for the manufacturer of a particular car. The user's information needs would be represented as `http://dbpedia.org.../Automobile` and `dbo:manufacturer`. The concept `Automobile` would be inferred by querying the linked data endpoint.

The contributions of this work are as follows:

1. A new perspective for analysing semantic query logs is provided.
2. A set of methods for extracting patterns in semantic query logs are described.
3. These methods are implemented in an interactive tool which enables the exploration of information needs revealed by the semantic query logs analysis.

The rest of this Section is structured as follows: Section 11.1.1 presents a review of the current state of the art in query logs analysis. Sections 11.1.2 and 11.1.3 discuss the approach followed in analysing query logs by modelling log entries and subsequent analysis results. Section 11.1.4 describes the dataset used for this analysis and finally, Section 11.1.5 presents the consumption of the analysis' results together with some observations.

11.1.1 Related Work

With the continuous growth of the Semantic Web, there has been a growing interest in studying different aspects related to its use and characteristics. The first large-scale

study was carried out by Ding and Fining in which they estimated the size of the SW to be approximately 300 million triples as of 2006.

Two recent studies [MMZ09, Hal09] tried to investigate whether casual Web users can satisfy answers to their information needs on the Semantic Web. The first study focused on extracting the main objects and attributes users were interested in from query logs which were then compared with Wikipedia templates to examine whether the schema of structured data on the web matches the users' needs as a key aspect to the success of semantic search. On the other hand, Halpin [Hal09] used a named entity recogniser to identify names of people and places together with WordNet [Fel98] to identify abstract concepts found in the users' queries. To investigate whether the Semantic Web contained answers to these queries, Falcon-S was used as a SW search engine and the results of executing the queries were analysed. On average, 1,339 URIs were returned for entity queries, while 26,294 URIs were returned for concept queries.

The work conducted by [MHCG10] is the first to study the usage of LOD. Unlike the previous studies which had a primary focus on the content of the queries, this study has a broader view of Web usage mining. It answers the questions of who is using LOD and how it is being used. The agents issuing the requests are classified into semantic and conventional based on their ability to process structured data. The two categories are further classified based on the agent type (e.g. bot, browser, etc...). Additionally, the study investigated the relevance of a dataset according to how its usage statistics are affected by events of public interest such as conferences or political events. The requests to a specific resource were measured around the time of occurrence of the associated real-time event to examine the influence on the access frequency. Similarly, [KKL11] defined a notion of relevance of a dataset or a particular Web resource after examining query logs provided by the USEWOD2011 data challenge[2].

The work done by [GFMPdlF11] builds on the work of [MHCG10] and performs further analysis on the nature of the SPARQL queries. The structure of the queries was examined to identify the most frequent pattern types, joins as well as SPARQL features such as *Optional* and *Union*. This information is valuable especially for the optimisation of SPARQL queries.

Although the previous studies took a step forwards towards mining/understanding the Web of Data, there is still scope for examining the content of queries and consuming it to understand trends and patterns in users' information needs.

11.1.2 Modelling Query Logs

In order to identify concepts and relations of interest from user queries, there is a need to formalise individual query logs to a structured and standardised representation. We propose the QLog (QueryLog) ontology to represent the main concepts and relations that can be extracted from a query log entry and by its subsequent analysis stages. The ontology has been developed by identifying the concepts of a log entry that follows the Combined Log Format (CLF)[3]. Figure 11.1 shows an example of a CLF log entry.

[2] http://data.semanticweb.org/usewod/2011/challenge.html
[3] http://httpd.apache.org/docs/1.3/logs.html#combined

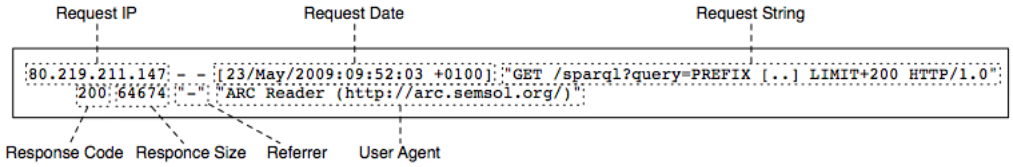


Figure 11.1: An example of a combined log format entry [MHCG10]

A query log entry is extracted to identify the different properties of the log entry - e.g. date and time, response size, response code, agent, query string (including SPARQL query) etc. In addition to the concepts which were identified from a CLF log entry, the QLog ontology also contains concepts to describe our analysis on the query log itself. The query string (identified as Request String in a CLF log entry) is further parsed and analysed to identify which concepts and relations have been queried for. The SPARQL query is also analysed to identify properties that can be derived, such as types and number of triple patterns, joins, filters and so on. Figure 11.2 (left block) shows the proposed QLog ontology, describing the CLF concepts as well as the analysed concepts (right block).

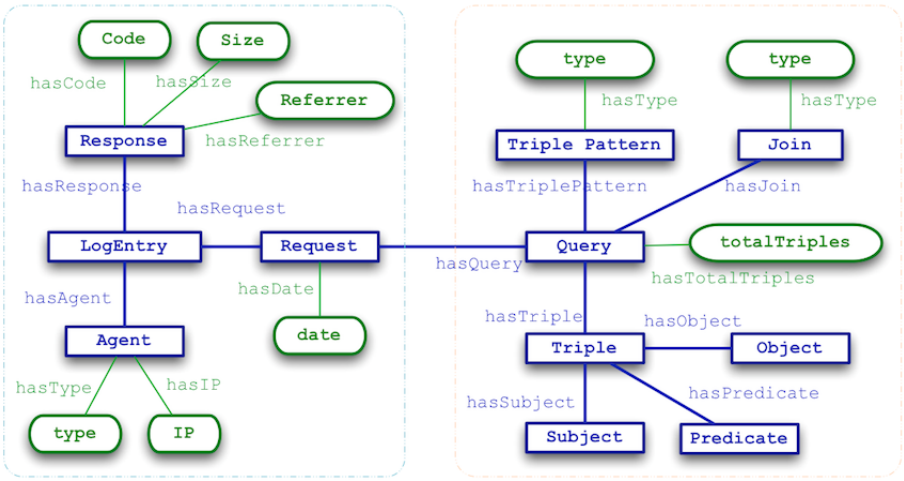


Figure 11.2: The Query Log (QLog) Ontology

It is important, however to understand that though there is scope for improving and engineering such ontologies, our main focus in this work is not to propose a highly engineered ontology to model query log entries, but use a formalised representation to structure extracted query log entries and query log analysis findings. This encourages and facilitates re-use and sharing the analysis with other interfaces and applications, as we discuss in the next sections.

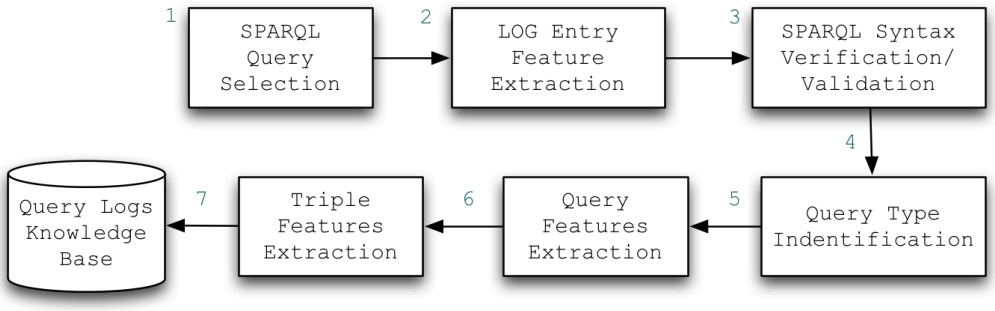


Figure 11.3: Query Logs analysis process diagram

11.1.3 Analysing Query Logs

Figure 11.3 shows the steps carried out in the analysis of the query logs. A Web server log is one or more files containing history of requests issued to a server. These requests are usually for different parts of a website and different kinds of information. For instance, a website such as `http://data.semanticweb.org` gets requests to particular web pages, RDF resources or to its SPARQL endpoint. Therefore, the first step in the analysis was to filter the dataset and extract the requests issued to the SPARQL endpoint. The properties associated with a log entry, as shown in the QLog ontology are extracted first. These include the agent type and IP address, the request date as well as the response code, referrer and result size. The IP address can be used in different user-based studies that require identifying requests coming from the same user. The request date was used in the study carried out by [KKL11] to investigate the relationship between LD resources and traffic of requests to these resources over different time windows. Agents requesting resources can be browsers (human usage), bots (machine usage), as well as tools (curl, wget, etc.) and data-services [MHCG10]. Identifying the kind of agents requesting resources and their distribution is useful for designers of LD tools to understand what information is being accessed and how.

The next step was to verify the correctness of each SPARQL query before extracting its properties. Queries producing parsing errors were excluded. For each successfully-parsed query, its type was first identified. The type can be either *Select, Ask, Construct* or *Describe*. In this analysis, we only considered the *Select* queries since it accounted for almost 97% of the query logs [GFMPdlF11]. A SPARQL query can have one or more triple patterns, joins, or solution modifiers such as *LIMIT* and *DISTINCT*, pattern matching constructs such as *OPTIONAL* and *UNION* as well as *FILTERs* for restricting the solution space. These different query parts are identified and triple patterns are analysed for extracting the properties associated with the query and the triples.

A triple pattern consists of three components; a *subject*, a *predicate* and an *object*. There are 8 types of triple patterns according to the place of existence of variables and constants. The most general one is <?S, ?P, ?O> which is used to retrieve everything in the queried data. More specific ones include patterns having one variable such as

<S, P, ?O> which retrieves the object values given a subject and a predicate, or two variables such as <S, ?P, ?O> retrieving all predicates and their values for a given subject. Finally the most specific triple pattern <S, P, O> does not ask for any data to be returned. After excluding the most general and specific triple patterns, the other types were identified when used in a query.

Each component in a triple pattern can be bound (having a specific value) or unbound (as a variable). Two triple patterns used in a query can be joined by using the same unbound component in both of them. For instance *?x hasName ?y* and *?x hasAge ?z* are joined using the unbound subject '?x'. Using this approach, four different join types were identified according to the place of the common variable in both patterns. For instance, the *Subject-Subject* join is one in which the common variable is found in the Subject place in both triple patterns. The other types are *Subject-Object, Predicate-Predicate* and *Object-Object*.

11.1.4 Dataset

DBpedia is the first dataset exposed on the Web as a result of a community effort to extract structured information from Wikipedia. Its knowledge base currently describes more than four million objects spanning multiple domains such as *People, Places* and *Species*. DBpedia is one of the largest datasets in the LOD cloud. In their study, Halpin et al.[Hal09] found that it dominated the results of queries (almost 83%) issued on the Semantic Web. It has been extensively used in other studies for different tasks [HMZ10]. To this end, an analysis done on DBpedia would be useful for the whole of LD and lessons learnt could be transferred to other datasets in the LOD cloud.

The data used in this study is made available by the USEWOD2011 data challenge[4]. The query logs follow the combined log format (shown earlier in Figure 11.1). The challenge data however included two additional fields, namely *Country code* and *Hash of original IP* to support both location and user-based analyses. The logs contained around five million queries issued to DBpedia over a time period of almost four months. Table 11.1 shows the basic statistics of the query logs.

In order to count the number of unique triple patterns used in the queries, the variables found in the patterns were first normalised. In that sense, the two triple

[4] http://data.semanticweb.org/usewod/2011/challenge.html

Table 11.1: Statistics summarising the query logs

Number of analysed queries	4951803
Number of unique triple patterns	2641098
Number of unique subjects	1168945
Number of unique predicates	2003
Number of unique objects	196221
Number of unique vocabularies	323

patterns '*...dbpedia...resource/X hasPage ?page*' and '*...dbpedia...resource/X ?hasPage ?homepage*' were considered to be similar since the same information is being requested. The large difference found in the number of unique subjects and objects matches the findings of [GFMPdlF11], as they showed that the most occurring triple pattern is <S P ?O>. This means that most of the queries require the value of the object, given a specific subject and predicate; the object is given as a variable and thus not counted.

In a similar way to analysing complexity of keyword queries on the Web of Documents in terms of query length, the first metric for Linked Data queries is the number of triple patterns used in a query. Almost 65% of the queries contained only one triple pattern, 18% contained two triple patterns while 15% contained three triple patterns. This shows that queries follow a power-law distribution in which most of the queries being simple and lie at the head of the distribution, while few more complicated queries with triple patterns ranging from 4 to 20 lie at the tail of the distribution. After excluding the most general and specific types of triple patterns (?S,?P,?O and S,P,O), the distribution of the other types is shown in Table 11.2.

The distribution shows that the most occurring triple pattern is <S P ?O>. This means that for almost 50% of the time, the information need is very specific - the value of a specific predicate for a given resource is required. The next most occurring type is <S ?P ?O>. Together, this means that for around 75% of the time, the information need is about a specific resource that the user of the query is interested in. Some Linked Data querying approaches build indexes to identify the relevant sources for answering a query or even use them to obtain the answer itself. In this sense, the identification of the most frequent triple patterns is valuable to optimise the equivalent indexes which in turn would improve the search performance.

In line with the above results, Table 11.2 shows that around 86% of the queries are simple with no joins. The number of joins then increase from 1 to 20 with an inverse relation to the percent of queries which decrease gradually. An interesting outcome of the analysis is that more than half of the joins (54%) were of type *Subject-Subject* and almost 32% were of type *Subject-Object*. Knowing this information is valuable for query planning and optimisation during the query execution process.

In addition to the basic graph patterns that can be found in a query, there are other

Table 11.2: Distribution of triple pattern and join types in the queries

TP Type	**Queries %**	**# Queries**	**# Joins**	**Queries %**	**# Queries**
S P ?O	49.55%	3760649	0	85.8%	4242899
S ?P ?O	25.94%	1968511	1	9.8%	485307
?S P ?O	12.84%	974882	2	2.6%	132128
?S P O	9.51%	722091	3	0.8%	37646
S ?P O	1.17%	88679	5	0.6%	30539
?S ?P O	0.97%	73888jj	6	0.07%	3560

pattern matching features that can be used such as *Optional, Union* and *Filter*. The **OPTIONAL** feature provides higher possibility to obtain results in a query. When used, it allows information to be returned if found, but also does not reject the solution when part of the query does not have matches in the data. Although this feature is useful while writing SPARQL queries, it is one, if not the most expensive operator in query evaluation as explained by [PAG09]. It is interesting to find that it occurred in only 15% of the queries. Although this might be good for query engines to avoid evaluating an expensive operator, it raises the question of why it is not used in LD queries. One explanation can be the knowledge and experience required to write similar queries.

UNION is used in SPARQL queries to allow combining graph patterns in the same way as does *OR* in SQL. The usage of union in the queries is limited to only 9.5%. Finally, the only feature that occurred in more than half of the queries (55%) is **FILTER**. It is used in SPARQL queries to restrict the results according to a given criteria. There are various expressions and operators that can be used together with a filter for different purposes. The most occurring expression in this analysis is *LANG* which restricts the results to the specified language.

The number of variables found in the select part of a SPARQL query shows how many data items the user is interested to see in the results. These can be instances, concepts or relations between them. This number was found in the analysis to range between 1 to 13 variables with 2 as the most used one followed by 1 then 3 select variables. Using select * in a query can be explained as either a lack of knowledge of the structure of the data or having a broad and non-specific information need; exploring the data. Interestingly, this accounted for 9.5% of the queries, which shows that in the other 90% of queries, users had more deterministic information need and knowledge of the data structure.

11.1.5 Visualisation of Query Logs

Analysis of query log entries can provide great insights into how individuals and software agents consume Linked Data. Making such analysis efforts available as formalised representation can be immensely valuable as this facilitates a generic approach to consume such data. For example, experts can now query such analysed data to gather an understanding of the information needs that emerge from the dataset. Visualisation tools and interfaces can consume such data thereby providing quick means to identify emerging trends and patterns from collective information needs. Figure 11.4 shows how we make use of our analysis to provide visualisations to users.

In order to consume the query log analysis findings, we have developed a software to visualise query log analysis data that has been formalised by the QLog ontology in the previous stage. The software presents two different types of information to the user:

1. Concept Graph - concepts in the data according to how often they have been queried (A, in Figure 11.4)

2. Predicate Sequence Tree - how users have queried for the data using predicate sequences (B, in Figure 11.4)

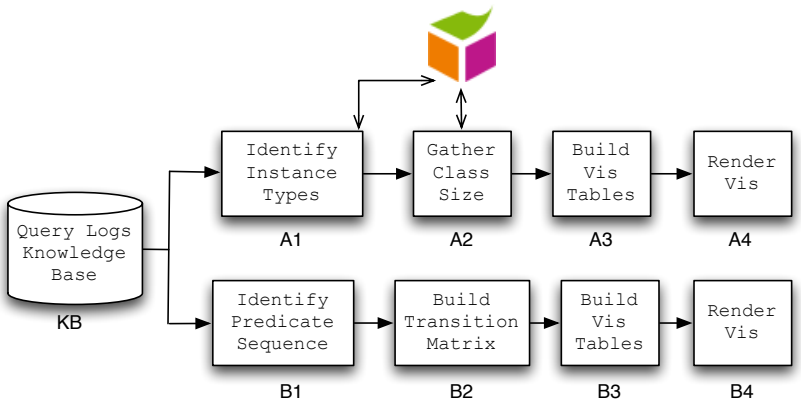


Figure 11.4: Consumption of Query Logs analysis results

The Query log analysis process described in Figure 11.3 results in RDF triples that are stored in a local triplestore (KB, Figure 11.4). In order to relate the information needs with concepts in the dataset, the Linked Data endpoint is initially queried to identify the types of instances being queried for (A1). For example, querying DBpedia endpoint for the type of the instance 'Acura ZDX' returns http://dbpedia.../Automobile. Once a type has been ascertained for a particular instance, the endpoint is queried again to understand how many instances in the data are that type (A2). In this example, DBpedia will be queried again for how many instances of Automobiles exist. This process would continue until all the instances and classes of the query logs have been analysed. The resulting information would then be assimilated into data tables (A3). A further interesting feature that can be identified by analysing SPARQL query logs is how users query for information especially when using multiple predicates to connect individual triple patterns. We refer to a predicate transition as a transition that occurs when the user's interest shifts from one predicate (in a triple pattern within a query) to another predicate (in the next triple pattern within the same query). Consider the following example:

```
PREFIX dbo: <http://dbpedia.org/ontology/>
SELECT ?name ?place WHERE {
        ?person dbo:birthPlace ?place.
        ?person foaf:name ?name.
}
```

Here, the user's predicate of interest transitions from *dbo:birthPlace* to *foaf:name*. In essence, the user is initially interested in looking at birthplaces of persons and then looking at their names. These transitions can now be collectively studied after analysing all of the formalised query logs. Studying such patterns can provide insights into how the user's information need shifts from one predicate to the next.

The process for visualising predicate sequences involves identifying the predicates that users have used to query. This can be retrieved by querying the KB for triple predicate instances, which provides the predicate sequences (B1). The triples are in-

stantiated according to the order they appeared in the query. This ensures that the consistency for the predicate sequence is maintained. The predicate sequences for all query log entries are then assimilated to construct a predicate transition matrix (B2), which is then converted to data tables (B3). The data tables thereby generated in A3 and B3 are then rendered (A4, B4) using the Prefuse Visualisation Toolkit[5].

11.1.6 SEMLEX - Exploring Information Needs

SEMLEX (SEMantic Logs EXplorer) was designed to explore and present such analysis on large query log datasets. The current implementation of SEMLEX provides users with two visualisations: *Concept Graph* and *Predicate Sequence Tree*. *Concept Graph* essentially explores the underlying ontologies, visually encoding nodes with information such as amount of data, query frequency and so on.

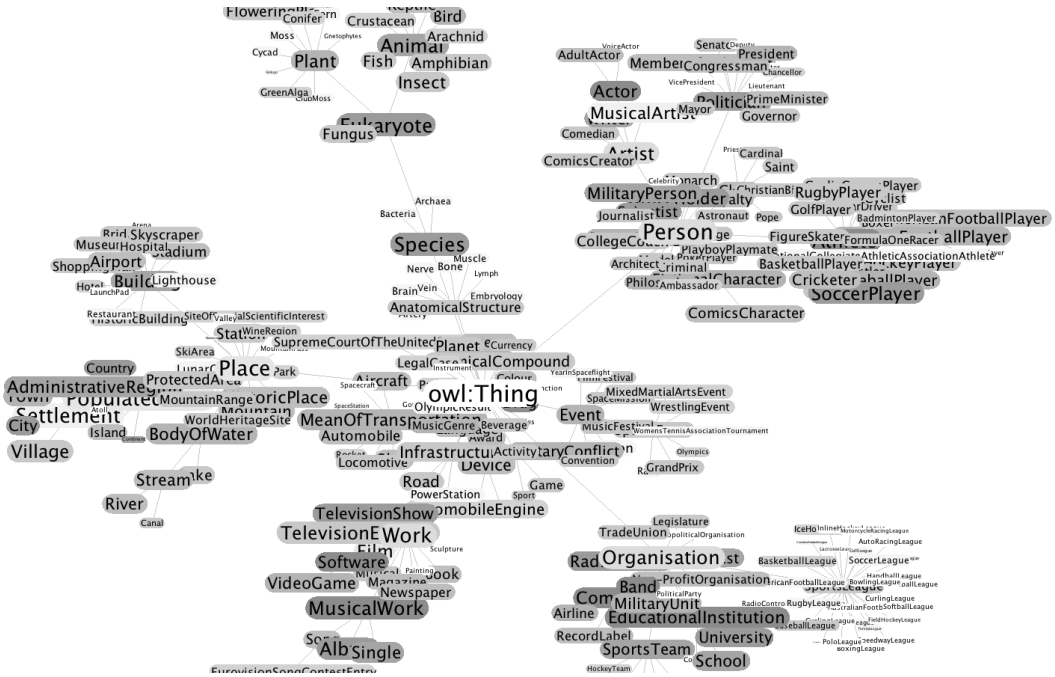


Figure 11.5: Exploring information needs of DBpedia users (Concept Graph). Node size represents the amount of instances (larger nodes represent more instances), colour represents the amount of user interest (darker nodes represent more interest)

Figure 11.5 shows a user exploring how DBpedia has been queried by users, with relation to the amount of data contained within the dataset. In this example, the ontology classes have been visually encoded using two sets of information - size (to represent how many instances are types of the concept within the dataset) and colour (to represent how many times the concept has been queried). The larger the size of a class, the greater the number of instances. Similarly, the darker the colour for a class, the more it has been queried for. For example, the concept 'wrestler' has been queried more times than a 'soccer player' (Figure 11.5, top right), even though the amount of

[5] http://prefuse.org/, Prefuse Visualisation Toolkit

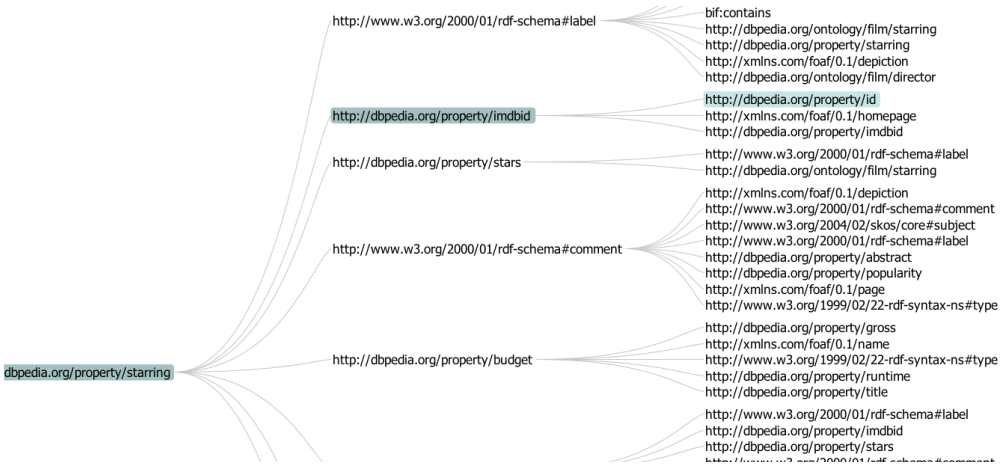


Figure 11.6: Exploring information needs of DBpedia users (Predicate Transition Tree). The figure shows that after cumulating predicate sequences of all the queries, for a particular property (e.g. dbprop:imdbid), what are the other predicates (e.g. dbprop:id, foaf:homepage, dbprop:imdbid, in descending order) used as the next predicate in one query.

instances for wrestlers is lesser than soccer players. While aggregating all the queries to identify which concepts are most queried for can provide an insight to data providers on which sections of an ontology are more 'interesting' to all users, it may be useful to explore how the users are querying the dataset. Regarding our case study, we found that the top concepts queried in DBpedia are: '*Person, Work, Organisation, Artist, Film, Place, PopulatedPlace, MusicalArtist, Settlement, Drug, Company, Software, Band, Actor, Athlete, MusicalWork, EducationalInstitution, Album, OfficeHolder, RadioStation, Country, Species, Politician, City, SoccerPlayer*'.

SEMLEX also enables users to see how predicates are connected to other predicates in individual queries. Clicking the 'PredicateSequence' tab loads the predicate sequence tree. The tool accumulates all the predicate transitions to build a transition matrix, which is then rendered as a tree. Figure 11.6 shows an example where a user explores the most used predicates that are associated with 'dbprop:starring'. The subtrees of the node are arranged according to their frequency of use - label being used most often while budget is less used. In our example, we focus on how users have queried for individuals who have starred in movies and then focus their search on IMDB entries. However, it seems that more users have looked for individuals who have starred in movies and then queried for the movies they have starred in or the movie directors. Observations such as this can be interesting for several other tools such as automatic query suggestions, recommender systems, search tools and so on.

Figure 11.7 shows the relation between the information found in the dataset (instances) and the distribution of queries requesting this information. The graph shows almost a direct correlation which we expected to find. Fortunately, this shows that users are querying more for concepts that have more information in the dataset or in other words, DBpedia is well structured towards information needs of LD users. However, the

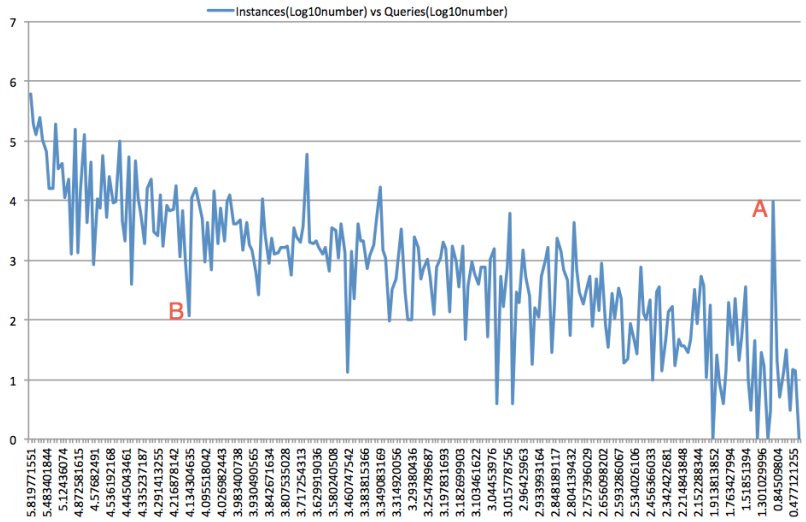


Figure 11.7: Distribution of queries against concepts instances size

graph showed some anomalies which were interesting to analyse. For instance, point A (shown in Figure 11.7) refers to the concept *Continent* in DBpedia which has only 10 instances, nevertheless being queried almost 10,000 times. Some of the concepts that had around similar number of instances to the frequency of queries are *PrimeMinister, Boxer, RecordLabel* and *Actor*. Others which had more information than they have been queried are *SoccerPlayer* which was queried 15193 times and has 65892 instance in the dataset and *Insect* which was queried around 1000 times and has 37742 instances. Finally, point B found on the other side of the graph, another anomaly arises for the concept *AutomobileEngine* which shows low interest while having sufficient amount of information. These results are in line with the ones illustrated in Figure 11.5.

Knowing such distribution and points of interests for users querying a dataset is of interest both to designers in terms of improving the structure of their data to better suit their users' information needs, and to consumers such as designers of semantic search and visualisation tools who can better support their users when they know more about their needs in advance.

11.2 Enriching Semantic Search Results using Query Logs

The difficulties in semantic search are not confined to abstraction, query construction or data visualisation. An additional problem focuses on the results of the query execution: what to return to the user and how to display it. Findings from the usability study presented earlier showed that semantic search tools should go a step further and augment the direct answer with associated information in order to provide a 'richer' experience for the user. Additionally, returning information related to entities and concepts found in a query might also be of interest to users [MMZ09, MBH+11]. Thus, this section

presents a proposal for a new approach which uses *collaborative knowledge* found in users' queries to help in addressing this problem.

The analysis of query logs presented in Section 11.1 showed that a small set of concepts and relations in a data set often account for a large proportion of the queries and thus may be of more interest to Linked Data users. Here, I extend this approach in order to demonstrate that careful log analysis can be viewed as a proxy for information needs and be used to enhance the search process at a number of different stages from query construction to results presentation. To achieve this, two models are used, each of which capture information regarding different aspects of the patterns present in the multi-user query logs.

The remainder of this Section is structured as follows. Section 11.2.1 describes the analysis performed on the semantic query logs and the models used to exploit this analysis. Subsequent sections show how these models can be used to address the problems described above. Section 11.2.3 shows how the results can be augmented in two different ways by exploiting the models. Section 11.2.4 illustrates how the models can be used to assist in visualising large data sets for query formulation. It should be emphasised that the details presented in the these sections are a proof of concept for the usage of the models presented in Section 11.2.2 (i.e. the results shown come from a "pen and paper" exercise as opposed to having been produced by an implementation of the approach).

11.2.1 Semantic Query Logs Analysis

Section 11.1 described a new approach for analysing and representing information needs using semantic query logs. Information needs was defined as "the set of concepts and properties users refer to while using SPARQL queries". A SPARQL query can have one or more triple patterns, solution modifiers (such as LIMIT), pattern matching constructs (such as OPTIONAL) and mechanisms for restricting the solution space (such as FILTERs). A triple pattern consists of three components: a subject, a predicate and an object with each component being either bound (having a specific value) or unbound (as a variable).

Extending on the analysis presented in Section 11.1, the information inherent in semantic query logs is formulated into two models which capture:

- the concepts used together in a query: the query-concepts model
- the predicates used with a concept: the concept-predicates model

The same extraction steps are followed but only triple patterns with bound subjects or objects are extracted to identify concepts (type of the subject/object) and predicates queried together which are used to build the proposed models. I used two sets of DBpedia query logs made available at the USEWOD[6] workshops (see Table 11.3). After extracting bound triple patterns (explained in Section 11.1.3), the types associated with each distinct resource appearing as a subject or an object in the query are identified by querying the Linked Data endpoint.

[6] http://data.semanticweb.org/usewod/2011(2012)/challenge.html

Table 11.3: Statistics summarising the query logs analysed.

	USEWOD2012	USEWOD2011
Number of queries	8866028	4951803
Number of unique triple patterns	4095011	2641098
Number of unique bound triple patterns	3619216	2571662

11.2.2 Models

In order to describe the proposed models, the following example query[7] is used throughout the rest of this section. The NL translation of this query is "what is the genre (e.g. 'Pop music') and instrument (e.g. keyboards) of the musician Ringo Starr?".

```
SELECT DISTINCT ?genre, ?instrument WHERE
{  <res:Ringo_Starr> ?rel <res:The_Beatles>.
   <res:Ringo_Starr> dbo:genre ?genre.
   <res:Ringo_Starr> dbo:instrument ?instrument.
}
```

Query-Concepts Model

This model captures the Linked Data concepts used in a whole query. All bound triple patterns (bound subject or object) in a single query are first identified and their types are retrieved from the Linked Data endpoint. The frequency of co-occurrence of each concept pair is accumulated. For the example query, the types retrieved for 'Ringo Starr' include `dbo:MusicalArtist` and `umbel:MusicalPerformer` while the 'The Beatles' has among its types `dbo:Band` and `schema:MusicGroup`. The frequency of co-occurrence of each concept in the first list with each concept in the second list is therefore incremented (e.g. `MusicalArtist` and `Band`).

Concept-Predicates Model

This model captures the Linked Data concepts and predicates in a query. Again, bound triple patterns are identified; however, only types of instances used as subjects are retrieved. The frequency of co-occurrence of each of the types with the predicate used in the triple pattern – if available – is accumulated. To illustrate, the second triple pattern in the example query increments the co-occurrence of `dbo:MusicalArtist` with `dbo:genre` and `umbel:MusicalPerformer` with `dbo:genre`.

11.2.3 Results Selection

In an attempt to improve the user experience, Google, Yahoo! and Bing use structured data embedded in web pages to enhance their search results (for example, by providing

[7] Prefix *res* refers to `http://dbpedia.org/resource/` and Prefix *dbo* refers to `http://dbpedia.org/ontology/`

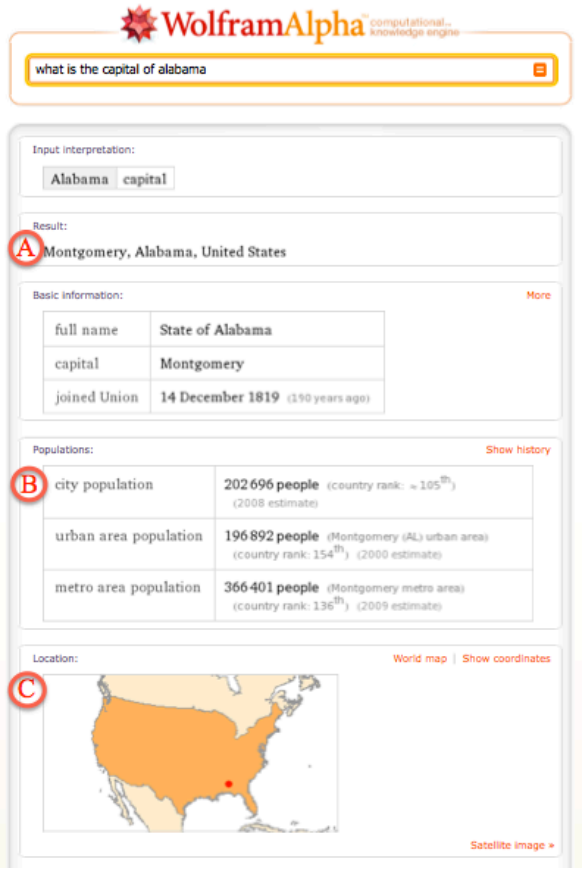


Figure 11.8: Results presentation in *WolframAlpha*. (A): natural language presentation of the answer; (B): population statistics; (C): map of the city.

supplementary information relevant to the query)[8]. WolframAlpha[9] is another example of systems providing more information together with the direct answer of a query. For example, as shown in Figure 11.8, the response to the query 'What is the capital of Alabama?' includes the natural language presentation of the answer (Figure 11.8, highlight A) as well as various population statistics (Figure 11.8, highlight B), a map showing the location of the city (Figure 11.8, highlight C), and other *related* information such as the current local time, weather and nearby cities.

Although Semantic Web search engines and question answering systems index much more structured data, a similar functionality (results enhancement) is not yet provided to their users. FalconS returns extra information together with each entity found as an answer to a query. It returns predicates associated with the entity in the underlying data (e.g. type, label, etc.); [WRE+10] showed that augmenting the answer with such extra information provides a richer user experience. This is, however, different from Linked Data mashups such as Sig.ma [TCC+10] and browsers such as Tabulator

[8]For example, Google Rich Snippets: http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html

[9]http://www.wolframalpha.com/

[BLCC+06] which attempt to create rich comprehensive views of entities and allow interactive exploration and navigation of Linked Data respectively. Furthermore, [MMZ09] and [MBH+11] suggested that returning information related to entities found in a query would be of interest to the user.

In an attempt to fill this gap, the rest of this section illustrates how the proposed models can be used to enrich results returned by semantic search systems. It distinguishes between providing more information about each result item and more information that is related to the query keywords including concepts and entities.

Return additional result-related information

To our knowledge, only VisiNav and FalconS return extra information about each entity in the result list. For the query "*What is the population of New York?*" and for the entity 'New York City', FalconS lists the following 10 properties with their values:

```
populationAsOf,dbprop:populationTotal,PopulatedPlace:populationTotal,
populationTotal,populationDensity,PopulatedPlace:populationDensity,
dbprop:populationDensitySqMi,dbprop:populationBlank,dbprop:populationMetro,
PopulatedPlace:populationUrban'
```

However, the strength of the proposed idea lies in utilising query logs as a source of collaborative knowledge able to capture perceptions of Linked Data entities and properties and use it to select which information to show the user rather than depending on a manually (or, indeed, randomly) predefined set. Additionally, [MMZ09, MBH+11] observed that a class of entities is usually queried with similar relations and concepts.

In order to return more information about each result item, the type of instance returned is first identified then the most frequently queried predicates associated with it are extracted from the *query-predicates* model. The top ranked ones are shown to the user, limited by the space available without cluttering the view and affecting the user experience. The user is given the ability to add more results which would retrieve the next set in the ranked list of predicates. Examples of concepts with their associated predicates list are given below:[10]

```
MusicalArtist-> rdfs:label,rdf:type,..,genre,associatedBand,occupation,
 instrument,birthDate,birthPlace,hometown,prop:yearsActive,foaf:surname,..
Film-> rdfs:label,rdf:type,..,prop:starring,prop:director,prop:name,release-
 Date,prop:gross,prop:budget,writer,producer,runtime,prop:cinematography,..
Country-> rdfs:label,rdf:type,..,capital,foaf:name,anthem,language,leader-
 Name,currency,largestCity,prop:areaKm,motto,prop:governmentType,..
```

[10] `prop` is used as a prefix for `http://dbpedia.org/property/` while the default prefix () is for `http://dbpedia.org/ontology/`

Return additional query-related information

Returning related information with the results of a query is an attempt to place the queried entities and concepts within context in the surrounding data which indeed assist users in discovering more information and useful findings that otherwise would not be noticed. Following our approach, the query concepts (include concepts and types of entities used in the query) are first identified. The most frequently occurring concepts used with them are extracted from the *query-concepts* model. Again, only a limited set (the actual size of which is determined on an application requirements basis) from the top ranked ones is returned. A set of examples are listed below with their co-occurring concepts.

```
MusicalArtist-> Film,Work,Band,Album,...,schema:Movie,MusicalWork,Place,
 Actor,TelevisionShow,WrittenWork,Model,City,Writer,schema:Event,..
City-> Book,Town,WorldHeritageSite,...,foaf:Person,Country,Organisation,
 SportsTeam,Scientist,Artist,Film,RadioStation,University,River,Hospital,..
Company->RecordLabel,foaf:Person,Work,...,LawFirm,Place,Software,Website,
 Broadcaster,TelevisionStation,Country,GovernmentAgency,Magazine,Convention,..
```

11.2.4 Data Visualisation

On the Semantic Web, supporting query formulation is provided by *view-based/visual-query interfaces* [BKGK05, CD11] which allow users to explore the underlying data. This can be very helpful for users, especially those unfamiliar with the search domain. A problem facing these tools is the technical limitations such as the number of items that can be included in a graph without cluttering the view and affecting user experience. This increases in heterogenous spaces like the open web since it is a challenge to decide what should be shown to users.

In an attempt to tackle this challenge and to identify a specific area of interest, [CD11] introduces a "specific-to-general" interface where it starts from an entity or a term entered by the user and builds a related subgraph extracted from the underlying data. After the user disambiguates the query term from a list of candidates, the tool returns a list of triples containing that term for the user to select from and add to his specific subgraph of data. In a dataset such as DBpedia – currently used by the tool's demo – this list will often contain thousands of triples for the user to examine in order to select the required ones.

Similarly, in my view, the *concepts-predicate* and *query-concepts* models presented above could be used to provide a more specific subgraph that allows users to explore the data around the entities they start with. This approach would be exploiting the collaborative knowledge collected from different users and applications to derive the selection of concepts and predicates added to the subgraph of interest. Using `Egypt` as a starting entity, Figure 11.9 shows a set of concepts and predicates associated with this entity's type – found in the models. Selecting a related concept retrieves a similar subgraph for the new one and shows the predicates connecting the two concepts.

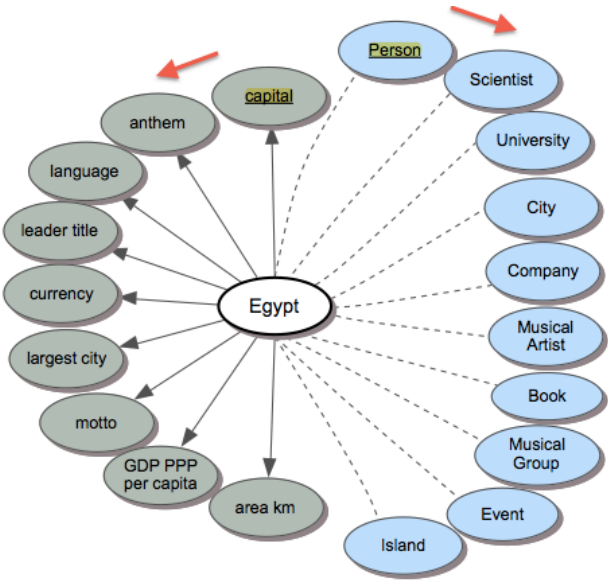


Figure 11.9: Results returned by the proposed approach for `Egypt`. Related concepts are on the right side and predicates on the left. For each side, elements are ranked with the top-most being most common and reducing in frequency in the direction of the arrows.

11.3 Response Time

An important requirement and current challenge for semantic search systems is with respect to the response time. Casual users are used to the speed of commercial search engines such as Google and Yahoo! which return results within fractions of a second. Therefore, semantic search should be able to provide similar performance, since users compare it to traditional search and are thus reluctant to wait for several seconds or few minutes for the results of their queries. While data warehousing can provide excellent query response times, it is limited in providing up-to-date data. I believe it is necessary to understand how to combine the best of the two worlds, and achieve the completeness and real-time response offered by having a data warehouse while balancing that with identifying new data on-the-fly [BHBL09]. Fortunately, the presentation of intermediate or partially-complete results – adopted by Sig.ma – can help reduce the perceived delay associated with the complete result set. Although only partial results are available initially, it both provides feedback that the search is executing properly, and allows the user to start thinking about the content of the results before the complete set is ready. However, it ought to be noted that this approach may induce confusion in the user as the screen content may change rapidly for a number of seconds. Adequate feedback is essential even for systems which exhibit high performance and good response times. Delays may occur at a number of points in the search process and may be the result of influences beyond the developer's control (e.g. network communication delays).

11.4 Summary

This chapter has presented proposals for future work. The first is an approach for exploring the information needs of Linked Data users by analysing semantic query logs. A case study of this approach was described using DBpedia, in which more than five million queries issued to its endpoint were analysed. I believe that this study provides useful insights by highlighting patterns and trends inherent in users' queries which indeed could be beneficial for different applications consuming Linked Data. In my view, different ways to extend this work include applying the same approach to examine other datasets with different features, such as the SWDogFood as a domain-specific dataset targeting Semantic Web researchers, as well as studying query logs that span multiple datasets such as the ones in the Linked Open Data Cloud Cache. The latter could present a more representative view of Linked Data queries in terms of size and domain coverage. Additionally, it could show how the query exchange between different datasets in the cloud occurs and whether the Linked Data principle of connecting datasets is being used in real-world queries. Finally, the analysis of these query logs could also be used for various tasks within different applications such as query completion suggestions or results enrichment within a semantic search tool. The latter is the second proposal presented in this chapter.

Following the wisdom of the crowd and exploiting collaborative knowledge found in these semantic query logs, the proposed approach attempts to create models of usage of Linked Data concepts and properties, in order to use them in results enrichment. As a proof of concept, I analysed more than 13 million DBpedia queries to create a sample of these models, which were then used in proposing a technique to provide more information about a result item as well as related information. Preliminary results have shown the potential of adopting the proposed models for improving semantic search results as well as query construction through data visualisation. To extend this work, it is important to evaluate the approach with respect to its performance as well as the quality and relevance of the returned results as perceived by real users. Although the current approach is promising, it would also be interesting to investigate the potential benefits of combining the current models with ones created from traditional query logs as opposed to semantic ones.

Additionally, my findings from the conducted evaluations showed the effect of results presentation on the user's experience and satisfaction. Small details such as organising answers in a table or having a visually-appealing display have a direct impact on readability and clarity of results and, in turn, user satisfaction. Thus, I believe, in addition to the above proposals, there is a need for more research and development focused on results generation and presentation to address these requirements (such as data cleaning, results filtering and management, and improved presentation formats). Ultimately, these are only a few examples of the tremendous opportunities offered by the Semantic Web in changing the way search is done.

Appendices

Appendix A

Assessing Usability of Semantic Search Query Approaches

This section includes experiment instructions and questionnaires given to the subjects in the usability study described in Chapter 7. Figures A.1 and A.2 show the instructions sheet provided for the subjects before starting the evaluation. Figures A.3 and A.4 show the *System Usability Scale (SUS) questionnaire* answered by each participant after evaluating each tool. Figures A.5 and A.6 show the *Extended questionnaire*, similarly answered after evaluating each tool, and designed to capture further aspects of the users' satisfaction with respect to the tool's query language and also the content returned in the results as well as how it was presented. Finally, Figures A.7 and A.8 show the *Demographics questionnaire* used to collect data such as age, profession and knowledge of linguistics, among others.

EVALUATION AND COMPARISON OF SEMANTIC SEARCH QUERY APPROACHES: INSTRUCTIONS DOCUMENT

You are invited to take part in a user study. Before you decide it is important for you to understand why the research is being done and what it will involve. Please take some time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like to have more information. Take your time to decide whether or not you wish to take part. Thank you for reading this.

Thank you very much for your voluntary participation in this study. It will take up about one hour of your time. After the experiment you will be rewarded for your effort.

The experiment is focused on assessing the usability of different semantic search query approach. The goal of this study is to test the user satisfaction with the approach at hand. During this experiment it is most important to measure the efficiency and usability of the query approach in formulating queries and completing the given search tasks.

It is possible to suspend or abort the experiment at any point without consequences. Your email address will be collected both by the controller software and the online questionnaires in order to be able to associate your results with your questionnaires. However, all information and data that will be collected during this experiment is of course strictly confidential. It will only be used for scientific purposes. It will be saved in an encoded and password protected form. (The data will further not be passed on to any third party.)
Further, we want to emphasize that any kind of critic is welcome and encouraged.

YOUR TASK

You will test five semantic search tools adopting different query approaches by rephrasing a list of natural language question fragments with varying complexity into each tool's query interface using its query language. You will be given a hands-on demo on the use of the tool and its query language then you will be asked to input the rephrased questions into the tools' interfaces.

The order of the experiment is as follows:

At first, you will be asked to read this instructions form. Secondly, you will be presented with a list of questions in English that you will be asked to reformulate into each tool's specific query language and input into the tool's interface. Beware: You do not have to note down the answers to the questions. After the practical part of the experiment you will be asked to answer three short questionnaires about your knowledge in specific aspects (such as Semantic Web and ontologies),

Figure A.1: Experiment instructions sheet provided for subjects in the usability study in Chapter 7.

demographics, satisfaction with the tools and opinion about the tools' usability.

The test leader can be consulted in case of questions and problems at any time, related to problems with completing the experiment, as opposed to help on answering the evaluation questions. Therefore, we encourage you to solve the tasks independently.

Please keep in mind that we do not intend to test your but the tool's abilities. We actually measure your satisfaction with the tool and not your behaviour.

Finally, note that you can skip a question whenever you feel the tool cannot find an answer or you are not satisfied with the results.

Figure A.2: Experiment instructions sheet provided for subjects in the usability study in Chapter 7.

I think I would like to use the system frequently. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the system unnecessarily complex. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought the system was easy to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I think I would need the support of a technical person to be able to use this system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the various functions in this system were well integrated. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I thought there was too much inconsistency in this system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure A.3: Post-search System Usability Scale (SUS) questionnaire presented in the usability study in Chapter 7.

I would imagine that most people would learn to use this system very quickly. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I found the system very tedious / troublesome to use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I felt very confident using the system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I needed to learn a lot of things before I could get going with this system. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure A.4: Post-search System Usability Scale (SUS) questionnaire presented in the usability study in Chapter 7.

I liked the presentation of the answers. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The system's query language was easy to understand and use. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I had the feeling that the system understood my questions correctly. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The information given in the answers was sufficient. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The feedback of the system was useful (I liked how it assisted me) *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

The system supported the functionality I expected. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

Figure A.5: Post-search Extended questionnaire presented in the usability study in Chapter 7.

I liked the system's response time. *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

I trust the system's answers *

	1	2	3	4	5	
Strongly Disagree	○	○	○	○	○	Strongly Agree

What did you like about the system you have just tested and why? *

How could the system you have just tested be improved to satisfy your needs and make it more attractive for you to use? *

Figure A.6: Post-search Extended questionnaire presented in the usability study in Chapter 7.

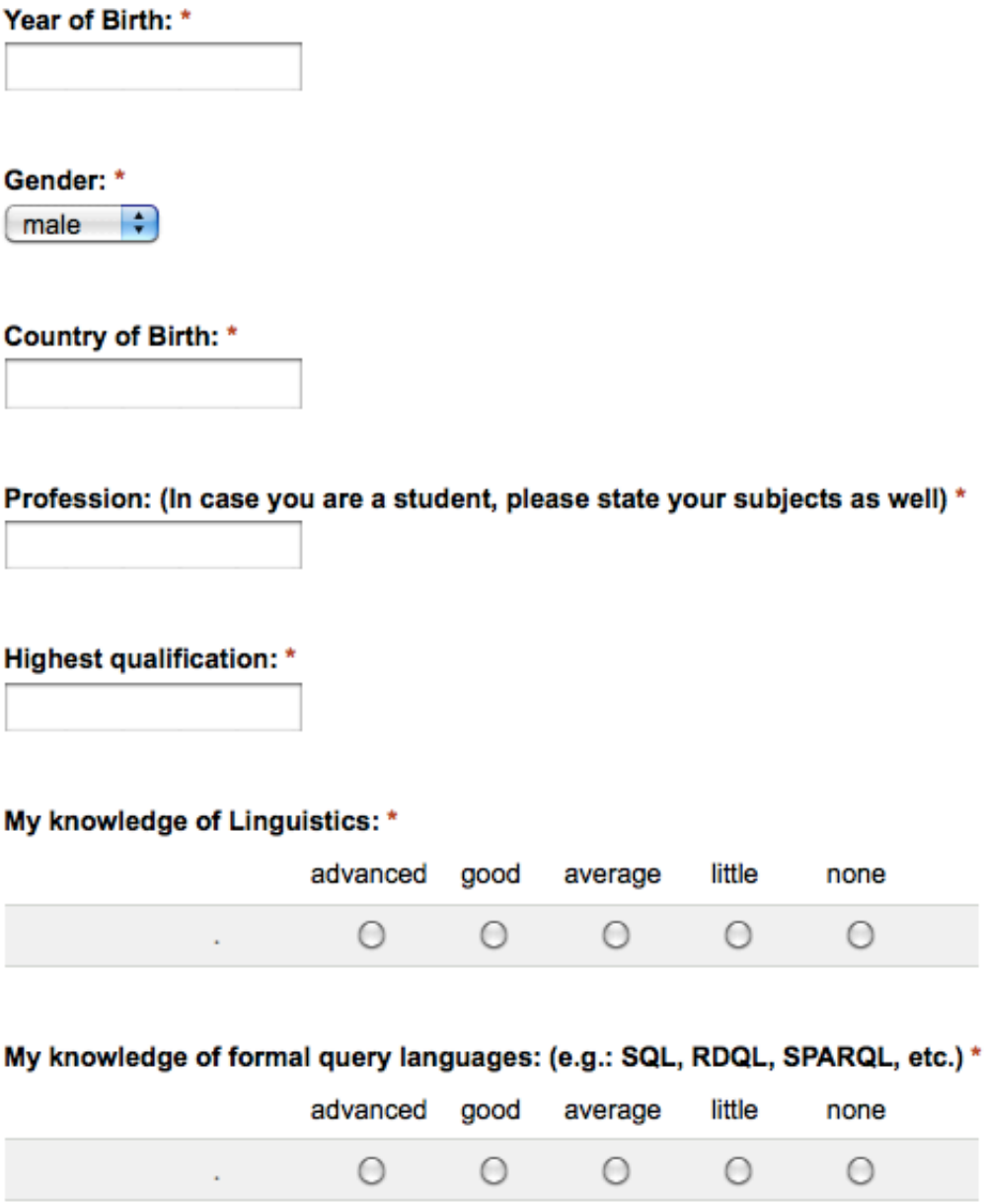


Figure A.7: Post-search Demographics questionnaire presented in the usability study in Chapter 7.

My knowledge of English: *

	mother tongue	advanced	good	average	little
.	○	○	○	○	○

My knowledge of Computers: *

	advanced	good	average	little	none
.	○	○	○	○	○

My knowledge of the Semantic Web: *

	advanced	good	average	little	none
.	○	○	○	○	○

My knowledge of Ontologies: *

	advanced	good	average	little	none
.	○	○	○	○	○

Do you regularly use a search engine (e.g. Google) *

yes

Figure A.8: Post-search Demographics questionnaire presented in the usability study in Chapter 7.

Appendix B

Evaluating Learnability of a Graph-based Query Approach

This section includes experiment instructions and questionnaires given to the subjects in the learnability study described in Chapter 8. Figures B.1 and B.2 show the instructions sheet provided for the subjects before starting the evaluation. The *System Usability Scale (SUS)* questionnaire presented above in Figures A.3 and A.4 is similarly used here. Additionally, Figures B.3 and B.4 show the *Extended questionnaire*, answered after each session and designed to include further questions related to the ease of use and learning of the interface as well as remembering how to use it. Finally, Figure B.5 shows the *Demographics questionnaire* used to collect data such as age, profession and knowledge of visual interfaces, among others.

EVALUATING LEARNABILITY OF A SEMANTIC SEARCH TOOL: INSTRUCTIONS DOCUMENT

You are invited to take part in a user study. Before you decide it is important for you to understand why the research is being done and what it will involve. Please take some time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like to have more information. Take your time to decide whether or not you wish to take part. Thank you for reading this.

Thank you very much for your voluntary participation in this study. It will take up about one hour of your time. After the experiment you will be rewarded for your effort.

The experiment is focused on assessing the usability and learnability of a semantic search query approach. The goal is to test the user efficiency and satisfaction with the approach at hand in completing the given search tasks. This will be conducted over a period of time to investigate the influence of practice and frequency of use on such aspects. Therefore, the experiment will take place over three sessions (during three consecutive days). The task itself will be almost the same in each session, with a change in the mode and amount of training provided as well as the evaluation questions.

It is possible to suspend or abort the experiment at any point without consequences. Your email address will be collected both by the controller software and the online questionnaires in order to be able to associate your results with your questionnaires. However, all information and data that will be collected during this experiment is of course strictly confidential. It will only be used for scientific purposes. It will be saved in an encoded and password protected form. (The data will further not be passed on to any third party.)
Further, we want to emphasize that any kind of critic is welcome and encouraged.

YOUR TASK
You will test a semantic search tool by rephrasing a list of natural language question fragments with varying complexity into the tool's query interface using its query language.

The order of the experiment is as follows:

At first, you will be asked to read this instructions form. Then for the first session, you will be given hands-on training on how to use the interface to formulate queries (with examples of complete search tasks). After this, you will be asked to formulate four questions in turn using the tool's interface. For the second and third sessions, the same process will be repeated. However, rather than training you

Figure B.1: Experiment instructions sheet provided for subjects in the learnability study in Chapter 8.

again, you will be shown best practices of using the interface, and common difficulties that were highlighted during the first session will be addressed. Additionally, you will be given time (equal to the training time) to practice using the interface and do any kind of queries you are interested in. Beware: You do not have to note down the answers to the questions. After this practical part of the experiment you will be asked to answer three short questionnaires about your knowledge in specific aspects (such as Semantic Web and ontologies), demographics, satisfaction with the tools and opinion about the tools' usability.

The test leader can be consulted in case of questions and problems at any time, related to problems with completing the experiment, as opposed to help on answering the evaluation questions. Therefore, we encourage you to solve the tasks independently.

Please keep in mind that we do not intend to test your but the tool's abilities. Finally, note that you can skip a question whenever you feel the tool cannot find an answer or you are not satisfied with the results.

Figure B.2: Experiment instructions sheet provided for subjects in the learnability study in Chapter 8.

The system's query language was easy to understand and use. *

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

Tasks can be performed in a straight-forward manner *

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

Exploring new features by trial and error is: *

	1	2	3	4	5	
Easy	○	○	○	○	○	Difficult

Remembering features and how to use them is: *

	1	2	3	4	5	
Easy	○	○	○	○	○	Difficult

Understanding the structure of the interface is: *

	1	2	3	4	5	
Easy	○	○	○	○	○	Difficult

Figure B.3: Post-search Extended questionnaire presented in the learnability study in Chapter 8.

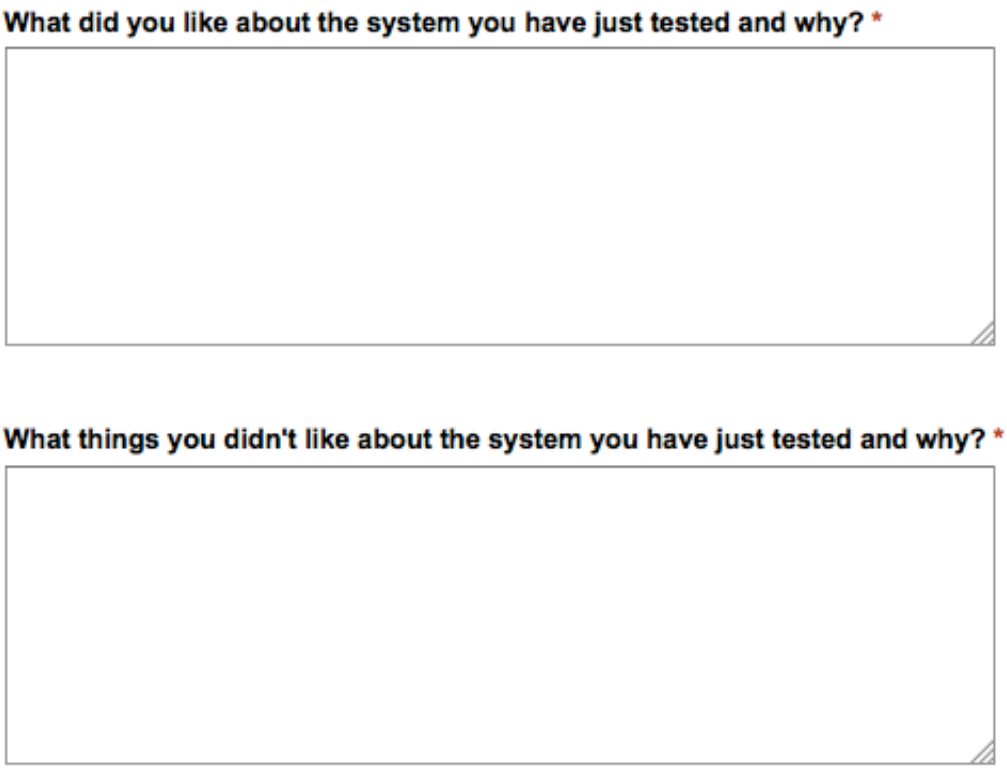


Figure B.4: Post-search Extended questionnaire presented in the learnability study in Chapter 8.

Year of Birth: *

Gender: *

male

Profession: (In case you are a student, please state your subjects as well) *

My knowledge of formal query languages: (e.g.: SQL, RDQL, SPARQL, etc.) *

	advanced	good	average	little	none
.	○	○	○	○	○

My knowledge of the Semantic Web: *

	advanced	good	average	little	none
.	○	○	○	○	○

My knowledge of Ontologies: *

	advanced	good	average	little	none
.	○	○	○	○	○

My knowledge of visual interfaces (any kinds of interface similar to the one you just tested) *

	advanced	good	average	little	none
.	○	○	○	○	○

Figure B.5: Post-search Demographics questionnaire presented in the learnability study in Chapter 8.

Appendix C

Evaluating the Hybrid Query Approach

This section includes experiment instructions and questionnaires given to the subjects in the evaluation of the hybrid approach described in Chapter 9. Figures C.1 and C.2 show the instructions sheet provided for the subjects before starting the evaluation. The *System Usability Scale (SUS)* questionnaire presented above is similarly used here. Additionally, Figure C.3 shows the *Extended questionnaire*, answered at the end of the experiment and designed to include a further question focused on the ease of use of the hybrid approach in addition to two open-ended questions. Finally, the same *Demographics questionnaire* shown above in Figure B.5 is used here.

EVALUATING USABILITY OF A NOVEL SEMANTIC SEARCH QUERY APPROACH: INSTRUCTIONS DOCUMENT

You are invited to take part in a user study. Before you decide it is important for you to understand why the research is being done and what it will involve. Please take some time to read the following information carefully and discuss it with others if you wish. Ask us if there is anything that is not clear or if you would like to have more information. Take your time to decide whether or not you wish to take part. Thank you for reading this.

Thank you very much for your voluntary participation in this study. It will take up about one hour of your time. After the experiment you will be rewarded for your effort.

The experiment is focused on assessing the usability of a novel semantic search query approach. During this experiment it is most important to measure the efficiency and usability of the query approach in formulating queries and completing the given search tasks.

It is possible to suspend or abort the experiment at any point without consequences. Your email address will be collected both by the controller software and the online questionnaires in order to be able to associate your results with your questionnaires. However, all information and data that will be collected during this experiment is of course strictly confidential. It will only be used for scientific purposes. It will be saved in an encoded and password protected form. (The data will further not be passed on to any third party.)
Further, we want to emphasize that any kind of critic is welcome and encouraged.

YOUR TASK

You will test a semantic search tool adopting the required query approach by rephrasing a list of natural language question fragments with varying complexity into the tool's query interface using its query language. You will be given a hands-on demo on the use of the tool and its query language then you will be asked to input the rephrased questions into the tools' interfaces.

The order of the experiment is as follows:

At first, you will be asked to read this instructions form. Secondly, you will be presented with a list of questions in English that you will be asked to reformulate into each tool's specific query language and input into the tool's interface. Beware: You do not have to note down the answers to the questions. After the practical part of the experiment you will be asked to answer three short questionnaires about your knowledge in specific aspects (such as Semantic Web and ontologies), demographics, satisfaction with the tools and opinion about the tools' usability.

Figure C.1: Experiment instructions sheet provided for subjects in the evaluation of the hybrid approach in Chapter 9.

The test leader can be consulted in case of questions and problems at any time, related to problems with completing the experiment, as opposed to help on answering the evaluation questions. Therefore, we encourage you to solve the tasks independently.

Please keep in mind that we do not intend to test your but the tool's abilities. We actually measure your satisfaction with the tool and not your behaviour.

Finally, note that you can skip a question whenever you feel the tool cannot find an answer or you are not satisfied with the results.

Figure C.2: Experiment instructions sheet provided for subjects in the evaluation of the hybrid approach in Chapter 9.

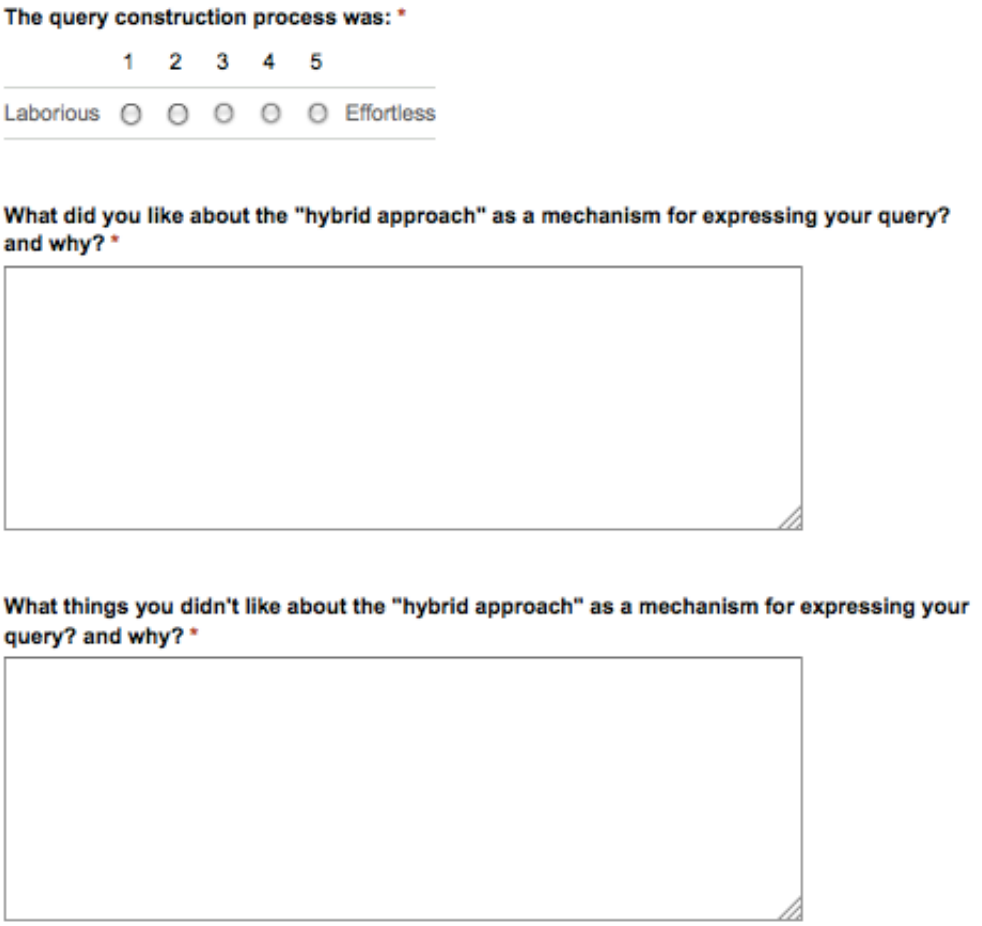


Figure C.3: Post-search Extended questionnaire presented in the evaluation of the hybrid approach in Chapter 9.

Bibliography

[AB05] Harith Alani and Christopher Brewster. Ontology ranking based on the analysis of concept structures. In *Proceedings of the 3rd international conference on Knowledge capture (K-CAP '05)*, 2005.

[ACHZ09] K. Alexander, R. Cyganiak, M. Hausenblas, and J. Zhao. Describing Linked Datasets - On the Design and Usage of voiD, the 'Vocabulary of Interlinked Datasets'. In *Proceedings of Linked Data on the Web Workshop (LDOW2009) at the 18th International World Wide Web Conference (WWW 2009)*, 2009.

[ADLS09] Eneko Agirre, Oier Lopez De Lacalle, and Aitor Soroa. Knowledge-based WSD on specific domains: performing better than generic supervised WSD. In *Proceedings of the 21st international joint conference on Artificial intelligence*, IJCAI'09, 2009.

[AG08] Renzo Angles and Claudio Gutierrez. The Expressive Power of SPARQL. In *Proceedings of the 7th International Conference on The Semantic Web (ISWC 2008)*, 2008.

[AKN+11] M-Dyaa Albakour, Udo Kruschwitz, Nikolaos Nanas, Yunhyong Kim, Dawei Song, Maria Fasli, and Anne De Roeck. AutoEval: an evaluation methodology for evaluating query suggestions using query logs. In *Proceedings of the 33rd European conference on Advances in information retrieval*, 2011.

[AM09] Omar Alonso and Stefano Mizzaro. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *Proceedings of SIGIR 2009 Workshop on The Future of IR Evaluation*, 2009.

[APY06] Javed A. Aslam, Virgil Pavlu, and Emine Yilmaz. A statistical method for system evaluation using incomplete judgments. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006)*. ACM, 2006.

[ATT10] W. Albert, T. Tullis, and D. Tedesco. *Beyond the Usability Lab: Conducting Large-Scale User Experience Studies*. Elsevier Science, 2010.

[Bai79] Lisanne Bainbridge. Verbal reports as evidence of the process operator's knowledge. *International Journal of Man-Machine Studies*, 11:411–436, 1979.

[BBFS05] James Bailey, François Bry, Tim Furche, and Sebastian Schaffert. *Web and Semantic Web Query Languages: A Survey*, volume 3564. Springer Berlin Heidelberg, 2005.

[BBG72] E.D. Barraclough, A.S. Barber, and W.A. Gray. *Medlars On-Line Search Formulation and Indexing*. University of Newcastle upon Tyne, Computing Laboratory, 1972.

[BCC+08] Ravish Bhagdev, Sam Chapman, Fabio Ciravegna, Vitaveska Lanfranchi, and Daniela Petrelli. Hybrid Search: Effectively Combining Keywords and Ontology-based Searches. In *Proceedings of the 5th European Semantic Web Conference (ESWC2008)*, 2008.

[BCH07] Christian Bizer, Richard Cyganiak, and Tom Heath. How to publish Linked Data on the Web. Web page, 2007. Revised 2008. Accessed 22/02/2010.

[BCYS07] Stefan Büttcher, Charles L. A. Clarke, Peter C. K. Yeung, and Ian Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2007)*, 2007.

[BDT83] Dina Bitton, David J. DeWitt, and Carolyn Turbyfill. Benchmarking Database Systems – A Systematic Approach. In *Proceedings of the 9th International Conference on Very Large Data Bases*, 1983.

[BH11] Sören Brunk and Philipp 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*, page 31, 2011.

[BHBL09] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5:1–22, 2009.

[BHH+13] Roi Blanco, Harry Halpin, Daniel M. Herzig, Peter Mika, Jeffrey Pound, Henry S. Thompson, and Thanh Tran. Repeatable and reliable semantic search evaluation. *Web Semantics*, 21:14–29, 2013.

[BI97] Pia Borlund and Peter Ingwersen. The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 53:225–250, 1997.

[BI98] Pia Borlund and Peter Ingwersen. Measures of relative relevance and ranked half-life: performance indicators for interactive IR. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[Bit85] D. Bitton et al. A Measure of Transaction Processing Power. *Datamation*, 31:112–118, 1985.

[BK03] Jeen Broekstra and Arjohn Kampman. SeRQL: A Second Generation RDF Query Language. In *Proceedings of SWAD-Europe Workshop on Semantic Web Storage and Retrieval*, 2003.

[BKGK05] Abraham Bernstein, Esther Kaufmann, Anne Göhring, and Christoph Kiefer. Querying Ontologies: A Controlled English Interface for End-users. In *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, 2005.

[BKK05] Abraham Bernstein, Esther Kaufmann, and Christian Kaiser. Querying the Semantic Web with Ginseng: A Guided Input Natural Language Search Engine. In *Proceedings of the 15th Workshop on Information Technology and Systems (WITS 2005)*, 2005.

[BKM09] A. Bangor, P. T. Kortum, and J. T. Miller. Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of Usability Studies*, 4:114–123, 2009.

[BL06] Tim Berners-Lee. Linked data design issues, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[BLCC+06] Tim Berners-Lee, Yuhsin Chen, Lydia Chilton, Dan Connolly, Ruth Dhanaraj, James Hollenbach, Adam Lerer, and David Sheets. Tabulator: Exploring and analyzing linked data on the semantic web. In *Proceedings of the 3rd International Semantic Web User Interaction Workshop*, 2006.

[BLF08] T. Berners-Lee and M. Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. Paw Prints, 2008.

[BLFM05] Tim Berners-Lee, Roy Thomas Fielding, and Larry Masinter. Uniform Resource Identifier (URI): Generic Syntax. *Network Working Group*, 66:1–61, 2005.

[BLK+09] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia - A crystallization point for the Web of Data. *Journal of Web Semantics*, 7:154–165, 2009.

[BM03] Mikhail Bilenko and Raymond J. Mooney. Employing Trainable String Similarity Metrics for Information Integration. In *Proceedings of the IJCAI-03 Workshop on Information Integration on the Web*, 2003.

[BMdR10] K. Balog, E. Meij, and M. de Rijke. Entity Search: Building Bridges between Two Worlds. In *Proceedings of Semantic Search 2010 Workshop at WWW 2010*, 2010.

[BMS07] J. Bhogal, A. Macfarlane, and P. Smith. A review of ontology based query expansion. *Information Processing & Management*, 43:866–886, 2007.

[BN09] Jens Bleiholder and Felix Naumann. Data fusion. *ACM Computing Surveys*, 41:1–41, 2009.

[Bor00] Pia Borlund. Experimental components for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 56:71–90, 2000.

[Bor09] P. Borlund. User-Centred Evaluation of Information Retrieval Systems. In A. Gker and Davies J., editors, *Information Retrieval: Searching in the 21st Century*, pages 21–37. John Wiley & Sons, 2009.

[BP02] Satanjeev Banerjee and Ted Pedersen. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, 2002.

[BP03] Satanjeev Banerjee and Ted Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.

[BRM04] Davide Buscaldi, Paolo Rosso, and Francesco Masulli. Integrating Conceptual Density with WordNet Domains and CALD Glosses for Noun Sense Disambiguation. In Jose Luis Vicedo, Patricio Martinez-Barco, Rafael Munoz, and Maximiliano Saiz Noeda, editors, *Advances in Natural Language Processing*, volume 3230, pages 183–194. Springer Berlin Heidelberg, 2004.

[Bro96] John Brooke. SUS: a quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor and Francis, 1996.

[Bro02] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[BRWC09] Abraham Bernstein, Dorothee Reinhard, Stuart Wrigley, and Fabio Ciravegna. SEALS Deliverable D13.1 Evaluation design and collection of test data for semantic search tools. Technical report, SEALS Consortium, November 2009.

[BS95] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, 1995.

[BS09] Christian Bizer and Andreas Schultz. The Berlin SPARQL Benchmark. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5:1–24, 2009.

[BSdV10] K. Balog, P. Serdyukov, and A.P. de Vries. Overview of the TREC 2010 Entity Track. In *TREC 2010 Working Notes*, 2010.

[BTC+08] Peter Bailey, Paul Thomas, Nick Craswell, Arjen P. De Vries, Ian Soboroff, and Emine Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674. ACM, 2008.

[BV85] N.J. Belkin and A. Vickery. *Interaction in Information Systems: A Review of Research from Document Retrieval to Knowledge-Based Systems*. British Library, 1985.

[BV00] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR2000)*, 2000.

[BV04] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2004)*, 2004.

[BWVS74] F.H. Barker, B.K. Wyatt, D.C. Veal, and United Kingdom Chemical Information Service. *Retrieval Experiments Based on Chemical Abstracts Condensates*. United Kingdom Chemical Information Service, 1974.

[CAC+12] E. Cabrio, A. Palmero Aprosio, J. Cojan, B. Magnini, F. Gandon, and A. Lavelli. QAKiS @ QALD-2. In *Proceedings of the Workshop on Interacting with Linked Data (ILD 2012) at the 9th Extended Semantic Web Conference (ESWC2012)*, 2012.

[CADW07] Charles L. A. Clarke, Eugene Agichtein, Susan Dumais, and Ryen W. White. The influence of caption features on clickthrough patterns in web search. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2007)*, 2007.

[Cat94] R. G. G. Cattell. An engineering database benchmark. In *Readings in database systems (2nd ed.)*. Morgan Kaufmann Publishers Inc., 1994.

[CCP+11] S. Campinas, D. Ceccarelli, T. E. Perry, R. Delbru, K. Balog, and G. Tummarello. The Sindice-2011 Dataset for Entity-Oriented Search in the Web of Data. In *Proceedings of the 1st International Workshop on Entity-Oriented Search (EOS)*, 2011.

[CD11] Aaron Clemmer and Stephen Davies. Smeagol: A Specific-to-General Semantic Web Query Interface Paradigm for Novices. In *Proceedings of the 22nd international conference on Database and expert systems applications (DEXA2011)*, 2011.

[CDBB09] Ping Chen, Wei Ding, Chris Bowes, and David Brown. A fully unsupervised word sense disambiguation method using dependency knowledge. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, 2009.

[CDKFZG06] Olivier Corby, Rose Dieng-Kuntz, Catherine Faron-Zucker, and Fabien Gandon. Searching the Semantic Web: Approximate Query Processing Based on Ontologies. *IEEE Intelligent Systems*, 21:20–27, 2006.

[CDN87] J.P. Chin, V.A. Diehl, and K.L. Norman. *Development of an Instrument Measuring User Satisfaction of the Human-computer Interface*. University of Maryland, 1987.

[CFMV11] Silvana Castano, Alfio Ferrara, Stefano Montanelli, and Gaia Varese. In Georgios Paliouras, Constantine D. Spyropoulos, and George Tsatsaronis, editors, *Knowledge-driven Multimedia Information Extraction and Ontology Evolution*, chapter Ontology and Instance Matching, pages 167–195. Springer-Verlag, 2011.

[CHHM07] Philipp Cimiano, Peter Haase, Jörg Heizmann, and Matthias Mantel. Orakel: A portable natural language interface to knowledge bases. Technical report, Institute AIFB, University of Karlsruhe, 2007.

[Chu11] Heting Chu. Factors affecting relevance judgment: a report from TREC Legal track. *Journal of Documentation*, 67:264 – 278, 2011.

[CK67] Carlos A. Cuadra and Robert V. Katter. Opening the black box of 'relevance'. *Journal of Documentation*, 23:291–303, 1967.

[CLCS92] Gregory A. Crawford, Arnold Lee, Lorene Connolly, and Y.L. Shylaja. OPAC user satisfaction and success: a study of four libraries. In *Proceedings of the 7th Conference on Integrated on-line Library Systems, IOLS 1992: Integrated on-line Library Systems*, 1992.

[Cle60] Cyril W. Cleverdon. Report on the first stage of an investigation onto the comparative efficiency of indexing systems. Technical report, The College of Aeronautics, Cranfield, England, 1960.

[Cle70] Cyril W. Cleverdon. The effect of variations in relevance assessments in comparative experimental tests of index languages. Technical report, The College of Aeronautics, Cranfield, England, 1970.

[Cle91] Cyril W. Cleverdon. The significance of the Cranfield tests on index languages. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1991)*, 1991.

[CLU+13] Philipp Cimiano, Vanessa López, Christina Unger, Elena Cabrio, Axel-Cyrille Ngonga Ngomo, and Sebastian Walter. Multilingual Question Answering over Linked Data (QALD-3): Lab Overview. In Pamela Forner, Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein, editors, *CLEF*, volume 8138 of *Lecture Notes in Computer Science*. Springer, 2013.

[CLY11] Vitor R. Carvalho, Matthew Lease, and Emine Yilmaz. Crowdsourcing for search evaluation. *SIGIR Forum*, 44:17–22, 2011.

[CMBT02] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[CMK66] C. W. Cleverdon, J. Mills, and M. Keen. Factors determining the performance of indexing systems. *Aslib Cranfield Research Project*, 1966.

[CMS09] Bruce Croft, Donald Metzler, and Trevor Strohman. *Search Engines: Information Retrieval in Practice*. Addison-Wesley Publishing Company, 2009.

[CMZG09] Olivier Chapelle, Donald Metlzer, Ya Zhang, and Pierre Grinspan. Expected reciprocal rank for graded relevance. In *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009.

[Coo68] William S. Cooper. Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, 19:30–41, 1968.

[Coo71] W. S. Cooper. A definition of relevance for information retrieval. *Information Storage and Retrieval*, 1971.

[Coo73a] W. S. Cooper. On selecting a measure of retrieval effectiveness. *Journal of the American Society for Information Science*, 24:87–100, 1973.

[Coo73b] W. S. Cooper. On selecting a measure of retrieval effectiveness part II. Implementation of the philosophy. *Journal of the American Society for Information Science*, 24:413–424, 1973.

[CPC98] Gordon V. Cormack, Christopher R. Palmer, and Charles L. A. Clarke. Efficient construction of large test collections. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1998)*, 1998.

[CPK+08] Ben Carterette, Virgil Pavlu, Evangelos Kanoulas, Javed A. Aslam, and James Allan. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2008)*, 2008.

[CR00] Claudio Carpineto and Giovanni Romano. Order-Theoretical Ranking. *Journal of the American Society for Information Science*, 51:587–601, 2000.

[CRBG02] Claudio Carpineto, Giovanni Romano, Fondazione Ugo Bordoni, and Vittorio Giannini. Improving retrieval feedback with multiple term-ranking function combination. *ACM Transactions on Information Systems*, 20:259–290, 2002.

[CRF03] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, 2003.

[CS13] P. Clough and M. Sanderson. Evaluating the performance of information retrieval systems using test collections. *Information Research*, 18(2):1–10, 2013.

[CSD+08] Richard Cyganiak, Holger Stenzhorn, Renaud Delbru, Stefan Decker, and Giovanni Tummarello. Semantic sitemaps: efficient and flexible access to datasets on the semantic web. In *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, 2008.

[CSH06] Namyoun Choi, Il-Yeol Song, and Hyoil Han. A survey on ontology mapping. *SIGMOD Rec.*, 35:34–41, 2006.

[CST+12] P. Clough, M. Sanderson, J. Tang, T. Gollins, and A. Warner. Examining the limits of crowdsourcing for relevance assessment. *Internet Computing, IEEE*, 17:32–38, 2012.

[Cua67] C.A. Cuadra. *Experimental Studies of Relevance Judgments. Final Report.* System Development Corporation, 1967.

[CWGQ08] Gong Cheng, Honghan Wu, Weiyi Ge, and Yuzhong Qu. Searching semantic web objects based on class hierarchies. In *Proceedings of of Linked Data on the Web Workshop (LDOW)*, 2008.

[CWNM02] Hang Cui, Ji-Rong Wen, Jian-Yun Nie, and Wei-Ying Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th international conference on World Wide Web (WWW '02)*, 2002.

[CZTR08] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *Proceedings of the international conference on Web search and web data mining*, 2008.

[DAC10] Danica Damljanovic, Milan Agatonovic, and Hamish Cunningham. Natural Language Interfaces to Ontologies: combining syntactic analysis and ontology-based lookup through the user interaction. In *Proceedings of the 7th Extended Semantic Web Conference (ESWC2010)*, 2010.

[dAM08] Brian de Alwis and Gail C. Murphy. Answering conceptual queries with Ferret. In *Proceedings of the 30th international conference on Software engineering (ICSE 2008)*, 2008.

[dBG+07] Mathieu d'Aquin, Claudio Baldassarre, Laurian Gridinoc, Sofia Angeletou, Marta Sabou, and Enrico Motta. Characterizing Knowledge on the Semantic Web with Watson. In *Proceedings of the 5th International EON Workshop: Evaluation of Ontologies and Ontology-Based Tools*, pages 1–10, 2007.

[DBW89] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. User acceptance of computer technology: a comparison of two theoretical models. *Manage. Sci.*, 35:982–1003, 1989.

[DFJ+04] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal C Doshi, and Joel Sachs. Swoogle: A Search and Metadata Engine for the Semantic Web. In *Proceedings of the 13th ACM Conference on Information and Knowledge Management*, 2004.

[Dic45] Lee Raymond Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, 1945.

[DLB00] Claude De Loupy and Patrice Bellot. Evaluation of document retrieval systems and query difficulty. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC 2000) Workshop*, 2000.

[DN09] Elena Demidova and Wolfgang Nejdl. Usability and Expressiveness in Database Keyword Search : Bridging the Gap. In *Proceedings of the PhD Workshop at the International Conference on Very Large Data Bases (VLDB 2009)*, 2009.

[DR07] Hong-Hai Do and Erhard Rahm. Matching large schemas: Approaches and evaluation. *Information Systems*, 32:857–885, September 2007.

[DR11] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: a survey. *Semantic Web Journal*, 2:89–124, 2011.

[Dra96] S. Draper. Overall task measurement and sub-task measurements. In *Proceedings of the 2nd Mira Workshop*, 1996.

[DS97] Xiaoying Dong and Louise T. Su. Search Engines on the World Wide Web and Information Retrieval from the Internet: A Review and Evaluation. *Online and CD-ROM Review*, 1997.

[dSSOH07] Roberto da Silva, Raquel Kolitski Stasiu, Viviane Moreira Orengo, and Carlos A. Heuser. Measuring quality of similarity functions in approximate data matching. *Journal of Informetrics*, 1:35–46, 2007.

[Dun96] M. D. Dunlop. Proceedings of the 2nd Mira Workshop. Research report tr-1997-2, University of Glasgow, 1996.

[Eis88] Michael B. Eisenberg. Measuring relevance judgments. *Information Processing & Management*, 24:373–389, 1988.

[EIV07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16, 2007.

[ES93] K A Ericsson and H A Simon. *Protocol analysis: Verbal reports as data*. MIT Press, 1993.

[ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[Fau03] L. Faulkner. Beyond the five-user assumption: Benefits of increased sample sizes in usability testing. *Behavior Research Methods, Instruments and Computers*, 35:379–383, 2003.

[FBY92] W.B. Frakes and R. Baeza-Yates. *Information retrieval: data structures & algorithms*. Prentice Hall, 1992.

[Fel98] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[FGKL02] Norbert Fuhr, Norbert Gövert, Gabriella Kazai, and Mounia Lalmas. INEX: INitiative for the Evaluation of XML Retrieval. *Proceedings of the SIGIR 2002 Workshop on XML and Information Retrieval*, 2006:1–9, 2002.

[FMS03] Kostas Fragos, Yannis Maistros, and Christos Skourlas. Word Sense Disambiguation using WordNet Relations. In *Proceedings of the 1st Balkan Conference in Informatics, Thessaloniki*, 2003.

[Fos72] D. J. Foskett. A note on the concept of relevance. *Information Storage and Retrieval*, 8:77–78, 1972.

[FS69a] Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[FS69b] Ivan P Fellegi and Alan B Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[GC07] M. Grubinger and P Clough. On the Creation of Query Topics for ImageCLEFphoto. In *Proceedings of the 3rd MUSCLE / ImageCLEF workshop on image and video retrieval evaluation*, 2007.

[GCV06] Julio Gonzalo, Paul Clough, and Alessandro Vallin. Overview of the CLEF 2005 interactive track. In *Proceedings of the 6th international conference on Cross-Language Evaluation Forum: accessing Multilingual Information Repositories*, 2006.

[GFA09] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: metrics, methodologies and guidelines. In *Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[GFMPdlF11] M. A Gallego, J. D Fernndez, M. A Martnez-Prieto, and P. de la Fuente. An Empirical Study of Real-World SPARQL Queries. *CoRR*, abs/1103.5043, 2011.

[GJG04] Laura A. Granka, Thorsten Joachims, and Geri Gay. Eye-tracking analysis of user behavior in WWW search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004.

[GN66] W. Goffman and V. A. Newill. A methodology for test and evaluation of information retrieval systems. *Information Storage and Retrieval*, 3:19–25, 1966.

[GN67] W. Goffman and V. A. Newill. Communication and epidemic processes. *Proceedings of the Royal Society*, 298(1454):316–334, 1967.

[GNC10] Michael Grubinger, Stefanie Nowak, and Paul Clough. Data Sets Created in ImageCLEF. In Henning Müller, Paul Clough, Thomas Deselaers, and Barbara Caputo, editors, *ImageCLEF*, volume 32 of *The Information Retrieval Series*, pages 19–43. Springer Berlin Heidelberg, 2010.

[GO04] Julio Gonzalo and Doug Oard. iCLEF 2004 track overview: Interactive Cross-Language Question Answering. In *Results of the CLEF 2004 Evaluation Campaign*, 2004.

[Gof64] William Goffman. On relevance as a measure. *Information Storage and Retrieval*, 2:201–203, 1964.

[GPH05] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3:158–182, 2005.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[GS11] Olaf Görlitz and Steffen Staab. SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions. In *Proceedings of the 2nd International Workshop on Consuming Linked Data at the 10th International Semantic Web Conference (ISWC2011)*, 2011.

[GY04] Fausto Giunchiglia and Mikalai Yatskevich. Element Level Semantic Matching. In *Proceedings of Meaning Coordination and Negotiation workshop at ISWC (MCN-04)*, 2004.

[Hal09] Harry Halpin. A Query-Driven Characterization of Linked Data. In *Proceedings of Linked Data on the Web Workshop (LDOW2009) at the 18th International World Wide Web Conference (WWW 2009)*, 2009.

[Har96] D Harper. User, task and domain. In *Proceedings of the 2nd Mira Workshop*, 1996.

[Har09] O. Hartig. Provenance Information in the Web of Data. In *Proceedings of Linked Data on the Web Workshop (LDOW2009) at the 18th International World Wide Web Conference (WWW 2009)*, 2009.

[Har11] Donna Harman. *Information Retrieval Evaluation*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2011.

[HBF09] Olaf Hartig, Christian Bizer, and Johann C. Freytag. Executing SPARQL Queries over the Web of Linked Data. In *Proceedings of the 8th International Semantic Web Conference (ISWC 2009)*, 2009.

[HC06] Lixin Han and Guihai Chen. The HWS hybrid web search. *Information and Software Technology*, 48:687 – 695, 2006.

[Hea09] M. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

[HEE+02] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the flow in web site search. *Commun. ACM*, 45:42–49, 2002.

[Her02] W.R. et al. Hersh. Factors associated with success in searching MEDLINE and applying evidence to answer clinical questions. *American Medical Informatics Association*, 9:283–293, 2002.

[HH97] S.P. Harter and C.A. Hert. Evaluation of Information Retrieval Systems: Approaches, Issues, and Methods. *Annual Review of Information Science and Technology (ARIST)*, 32:3–94, 1997.

[HHD+07] Andreas Harth, Aidan Hogan, Renaud Delbru, Jürgen Umbrich, Sen O'Riain, and Stefan Decker. SWSE: Answers Before Links! In *Semantic Web Challenge*, 2007.

[HHK+09] Andreas Harth, Katja Hose, Marcel Karnstedt, Axel Polleres, Kai-Uwe Sattler, and Jürgen Umbrich. On Lightweight Data Summaries for Optimised Query Processing over Linked Data. Technical report, DERI, NUI Galway, 2009.

[HHL+09] Philipp Heim, Sebastian Hellmann, Jens Lehmann, Steffen Lohmann, and Timo Stegemann. Relfinder: Revealing relationships in rdf knowledge bases. In *Semantic Multimedia*, pages 182–187. Springer, 2009.

[HHM+10] Harry Halpin, Daniel M Herzig, Peter Mika, Roi Blanco, Jeffrey Pound, Henry S Thompson, and Thanh Tran Duc. Evaluating Ad-Hoc Object Retrieval. In *Proceedings of the 1st International Workshop on Evaluation of Semantic Technologies (IWEST)*, 2010.

[Hil01] C.R. Hildreth. Accounting for users' inflated assessments of on-line catalogue search performance and usefulness: an experimental study. *Information Research: an international electronic journal*, 6(2), 2001.

[Hit79] E.E. Hitchingham. A Study of the Relationship between the Search Interview of the Intermediary Searcher and the Online System User, and the Assessment of Search Results as Judged by the User. Final Report. Research report, Oakland Univ., Rochester, MI. Kresge Library., 1979.

[HMZ10] Peter Haase, Tobias Mathäß, and Michael Ziller. An evaluation of approaches to federated query processing over linked data. In *Proceedings of the 6th International Conference on Semantic Systems (I-SEMANTICS 2010)*, 2010.

[HO99] William Hersh Hersh and Paul Over. TREC-9 Interactive Track Report. In *Proceedings of the Text REtrieval Conference (TREC-9)*, 1999.

[HO00] William Hersh and Paul Over. SIGIR workshop on interactive retrieval at TREC and beyond. *SIGIR Forum*, 34, 2000.

[HPUZ10] Aidan Hogan, Axel Polleres, Jürgen Umbrich, and Antoine Zimmermann. Some entities are more equal than others: statistical methods to consolidate Linked Data. In *Proceedings of the Workshop on New Forms of Reasoning for the Semantic Web: Scalable & Dynamic (NeFoRS2010)*, 2010.

[HS00] Christoph Hölscher and Gerhard Strube. Web search behavior of Internet experts and newbies. *Computer Networks*, 33:337–346, 2000.

[HTP+00] William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kramer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000.

[Hul93] David Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, 1993.

[HV03] Eero Hyvnen and Kim Viljanen. Ontogator: combining view- and ontology-based search with semantic browsing. In *Proceedings of XML*, 2003.

[HV08] S. Huuskonen and P. Vakkari. Students' search process and outcome in Medline in writing an essay for a class on evidence-based medicine. *Journal of Documentation*, 64:287–303, 2008.

[HY01] Ingrid Hsieh-Yee. Research on Web search behavior. *Library & Information Science Research*, 23:167–185, 2001.

[HY07] Orland Hoeber and Xue Dong Yang. User-Oriented Evaluation Methods for Interactive Web Search Interfaces. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops (WI-IATW 2007)*, 2007.

[HZ10] O. Hartig and J Zhao. Publishing and consuming provenance metadata on the Web of Linked Data. In *Proceedings of the Provenance and Annotation Workshop*, 2010.

[IJ05] Peter Ingwersen and Kalervo Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer, 2005.

[Ing92] P. Ingwersen. *Information retrieval interaction*. Taylor Graham, 1992.

[Inm05] W.H. Inmon. *Building the data warehouse*. Wiley, 2005.

[ISO98] ISO. ISO 9241-11:1998 Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Technical report, International Organization for Standardization, 1998.

[IV98] Nancy Ide and Jean Vronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24:1–40, 1998.

[JÏ1] Kalervo Järvelin. Evaluation. In I. Ruthven and D. Kelly, editors, *Interactive Information Seeking, Behaviour and Retrieval*, pages 113–138. Facet Publishing, 2011.

[Jan93] Joseph W. Janes. On the distribution of relevance judgements. In *Proceedings of the ASIS Annual Meeting*, 1993.

[JB77] K.S. Jones and R.G. Bates. *Report on a Design Study for the 'ideal' Information Retrieval Test Collection*. Computer Laboratory, University of Cambridge, 1977.

[JCW03] Christine Jenkins, Cynthia L. Corritore, and Susan Wiedenbeck. Patterns of information seeking on the Web: A qualitative Study of domain expertise and Web expertise. *IT and Society*, 1:64–89, 2003.

[Jen05] Judy Jeng. Usability assessment of academic digital libraries: effectiveness, efficiency, satisfaction, and learnability. *Libri: International Journal Of Libraries And Information Services*, 55(2-3):96–121, 2005.

[JFH98] Joemon M. Jose, Jonathan Furner, and David J. Harper. Spatial querying for image retrieval: a user-oriented evaluation. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.

[JGH03] F. C. Johnson, J. R. Griffiths, and R. J. Hartley. Task dimensions of user evaluations of information retrieval systems. *Information Research*, 8(4):8–4, 2003.

[JGP+05] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005)*, 2005.

[JHJ08] Hideo Joho, David Hannah, and Joemon M. Jose. Comparing collaborative and independent search in a recall-oriented task. In *Proceedings of the 2nd international symposium on Information interaction in context*, 2008.

[JHS+10] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology alignment for linked open data. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, 2010.

[JK00] Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2000)*, 2000.

[JK02] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20:422–446, 2002.

[Jon71] K.S. Jones. *Automatic keyword classification for information retrieval.* Archon Books, 1971.

[JRGH07] Frances Johnson Jillian R. Griffiths and Richard J. Hartley. User Satisfaction as a Measure of System Performance. *Journal of Librarianship and Information Science*, 39:142–152, 2007.

[JS05] Bernard J. Jansen and Amanda Spink. An analysis of Web searching by European AlltheWeb.com users. *Information Processing and Management: an International Journal*, 41(2):361–381, 2005.

[JSP05] Bernard J. Jansen, Amanda Spink, and Jan Pedersen. A temporal comparison of AltaVista Web searching: Research Articles. *Journal of the American Society for Information Science and Technology*, 56:559–570, April 2005.

[JVY+10] Prateek Jain, Kunal Verma, Peter Z. Yeh, Pascal Hitzler, and Amit P. Sheth. LOQUS: Linked Open Data SPARQL Querying System. Technical report, Kno.e.sis Center, Wright State University and Accenture Technology Labs, 2010.

[KA08] Mika Kaki and Anne Aula. Controlling the complexity in comparing search user interfaces via user studies. *Information Processing & Management*, 44:82–91, 2008.

[KA09] Evangelos Kanoulas and Javed A. Aslam. Empirical Justification of the Gain and Discount Function for nDCG. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, 2009.

[Kat68] R.V. Katter. The influence of scale form on relevance judgments. *Information Storage and Retrieval*, 4:1–11, 1968.

[Kau07] Esther Kaufmann. *Talking to the Semantic Web — Natural Language Query Interfaces for Casual End-Users*. PhD thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, September 2007.

[Kaz11] Gabriella Kazai. In search of quality in crowdsourcing for search engine evaluation. In *Proceedings of the 33rd European conference on Advances in information retrieval*, 2011.

[KB96] Jürgen Koenemann and Nicholas J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 1996)*, 1996.

[KB07] Esther Kaufmann and Abraham 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 2007)*, 2007.

[KB10] Esther Kaufmann and Abraham Bernstein. Evaluating the Usability of Natural Language Query Languages and Interfaces to Semantic Web Knowledge Bases. *Journal of Web Semantics*, 8:377–393, 2010.

[KBLP55] Allen Kent, Madeline M. Berry, Fred U. Luehrs, and J. W. Perry. Machine literature searching VIII. Operational criteria for designing information retrieval systems. *American Documentation*, 6:93–101, 1955.

[KBZ06] Esther Kaufmann, Abraham Bernstein, and Renato Zumstein. Querix: A Natural Language Interface to Query Ontologies Based on Clarification Dialogs. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, 2006.

[KD72] E. M. Keen and J. A. Digger. Report of an information Science Index Languages Test. *Aberystwyth, Department of Information Retrieval Studies, College of Librarianship Wales*, 1972.

[Kel09] Diane Kelly. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval*, 3:1–224, 2009.

[Kem74] D.A. Kemp. Relevance, pertinence and information system development. *Information Storage and Retrieval*, 10:37 – 47, 1974.

[KHZ08] Kenneth A. Kinney, Scott B. Huffman, and Juting Zhai. How evaluator domain expertise affects search result relevance judgments. In *Proceedings of the 17th ACM conference on Information and knowledge management (CIKM 2008)*. ACM, 2008.

[KKL11] Markus Kirchberg, Ryan K. L. Ko, and Bu Sung Lee. From Linked Data to Relevant Data – Time is the Essence. *CoRR*, abs/1103.5046, 2011.

[KL07] Diane Kelly and Jimmy Lin. Overview of the TREC 2006 ciQA task. *SIGIR Forum*, 41:107–116, 2007.

[KM03] Dan Klein and Christopher D. Manning. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems*, 2003.

[KMA+03] G. Karvounarakis, A. Magganaraki, S. Alexaki, V. Christophides, D. Plexousakis, Michel Scholl, and Karsten Tolle. Querying the Semantic Web with RQL. *Computer Networks*, 42:617–640, 2003.

[KPT+04] Atanas Kiryakov, Borislav Popov, Ivan Terziev, Dimitar Manov, and Damyan Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79, 2004.

[KR02] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.* John Wiley & Sons, Inc., 2002.

[KR10] Hanna Köpcke and Erhard Rahm. Frameworks for entity matching: A comparison. *Data & Knowledge Engineering*, 69(2):197–210, 2010.

[KS99] Atanas Kiryakov and Kiril Simov. Ontologically Supported Semantic Matching. In *Proceedings of NODALIDA99: Nordic Conference on Computational Linguistics*, pages 9–10, 1999.

[KV00] Paul B. Kantor and Ellen M. Voorhees. The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. *Information Retrieval*, 2:165–176, 2000.

[Lan68] F.W. Lancaster. *Evaluation of the MEDLARS demand search service: by F. W. Lancaster.* U.S. Dept. of Health, Education, and Welfare, Public Health Service, 1968.

[Les86] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, 1986.

[Lew94] James R. Lewis. Sample Sizes for Usability Studies: Additional Considerations. *The Journal of the Human Factors and Ergonomics Society*, 36:368–378, 1994.

[Lew95] James R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7:57–78, 1995.

[LFMS12] Vanessa López, Miriam Fernández, Enrico Motta, and Nico Stieler. PowerAqua: supporting users in querying and exploring the semantic web. *Semantic Web Journal*, 3:249–265, 2012.

[LM07] Serge Linckels and Christoph Meinel. Semantic Interpretation of Natural Language User Input to Improve Search in Multimedia Knowledge Base. *Information Technology*, 1(49):40–48, 2007.

[LMU06] Vanessa López, Enrico Motta, and Victoria Uren. PowerAqua: Fishing the Semantic Web. In *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 393–410. Springer Berlin Heidelberg, 2006.

[LMUS07] Vanessa López, Enrico Motta, Victoria Uren, and Marta Sabou. Literature review and state of the art on Semantic Question Answering, 2007.

[Los98] Robert M. Losee. *Text retrieval and filtering: analytic models of performance.* Kluwer Academic Publishers, 1998.

[LP98] Andreas Lecerof and Fabio Paternò. Automatic Support for Usability Evaluation. *IEEE Transactions on Software Engineering*, 24:863–888, 1998.

[LPM05] Vanessa López, Michele Pasin, and Enrico Motta. AquaLog: An Ontology-Portable Question Answering System for the Semantic Web. In *The Semantic Web: Research and Applications*. Springer, 2005.

[LUCM13] Vanessa López, Christina Unger, Philipp Cimiano, and Enrico Motta. Evaluating question answering over linked data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 21:3 – 13, 2013.

[LUM06] Yuangui Lei, Victoria Uren, and Enrico Motta. SemSearch: A Search Engine for the Semantic Web. In *Managing Knowledge in a World of Networks*, volume 4248 of *Lecture Notes in Computer Science*, pages 238–245. Springer Berlin Heidelberg, 2006.

[LUSM11] Vanessa López, Victoria Uren, Marta Sabou, and Enrico Motta. Is question answering fit for the Semantic Web? A survey. *Semantic Web*, 2:125–155, 2011.

[MÏ0] Henning Müller. Creating Realistic Topics for Image Retrieval Evaluation. In Henning Müller, Paul Clough, Thomas Deselaers, and Barbara Caputo, editors, *ImageCLEF*, volume 32 of *The Information Retrieval Series*, pages 45–61. Springer Berlin Heidelberg, 2010.

[Mäk06] Eetu Mäkelä. *View-based search interfaces for the semantic web*. PhD thesis, University of Helsinki, 2006.

[MBF+90] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[MBH+09] Edgar Meij, Marc Bron, Laura Hollink, Bouke Huurnink, and Maarten Rijke. Learning Semantic Query Suggestions. In *Proceedings of the 8th International Semantic Web Conference (ISWC2009)*, 2009.

[MBH+11] Edgar Meij, Marc Bron, Laura Hollink, Bouke Huurnink, and Maarten de Rijke. Mapping queries to the Linking Open Data cloud: A case study using DBpedia. *Journal of Web Semantics*, 9:418 – 433, 2011.

[McG95] Joseph E. McGrath. Methodology matters: doing research in the behavioral and social sciences. In *Human-computer interaction*, pages 152–169. Morgan Kaufmann Publishers Inc., 1995.

[ME96] Alvaro Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[MGSS10] Ian Millard, Hugh Glaser, Manuel Salvadores, and Nigel Shadbolt. Consuming multiple linked data sources: Challenges and Experiences. In *Proceedings of the 1st International Workshop on Consuming Linked Data (COLD2010)*, 2010.

[MHCG10] Knud Möller, Michael Hausenblas, Richard Cyganiak, and Gunnar Aastrand Grimnes. Learning from Linked Open Data Usage: Patterns and Metrics. In *Proceedings of the WebSci10: Extending the Frontiers of Society On-Line*, 2010.

[Miz98] Stefano Mizzaro. How many relevances in information retrieval? *Interacting with Computers*, 10:305–322, 1998.

[MM04] Frank Manola and Eric Miller, editors. *RDF Primer*. W3C Recommendation. World Wide Web Consortium, February 2004.

[MMZ09] E. Meij, P. Mika, and H. Zaragoza. Investigating the Demand Side of Semantic Search through Query Log Analysis. In *Proceedings of the 2nd International Semantic Search Workshop (SemSearch 2009)*, 2009.

[MND+12] Alberto Musetti, Andrea Giovanni Nuzzolese, Francesco Draicchio, Valentina Presutti, Eva Blomqvist, Aldo Gangemi, and Paolo Ciancarini. Aemoo: Exploratory search based on knowledge patterns over the semantic web. *Semantic Web Challenge*, 2012.

[MRS08] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MRV+03] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesus Herrera, Anselmo Penas, Victor Peinado, Felisa Verdejo, Maarten de Rijke, and Ro Vallin. The Multiple Language Question Answering Track at CLEF 2003. In *CLEF 2003 Workshop*, 2003.

[MSB98] Mandar Mitra, Amit Singhal, and Chris Buckley. Improving automatic query expansion. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1998)*, 1998.

[MSS03] Bernardo Magnini, Luciano Serafini, and Manuela Speranza. Making Explicit the Semantics Hidden in Schema Models. In *Proceedings of the Workshop on Human Language Technology for the Semantic Web and Web Services, held at ISWC-2003*, 2003.

[MSSR02] Libby Miller, Andy Seaborne, Andy Seaborne, and Alberto Reggiori. Three Implementations of SquishQL, a Simple RDF Query Language. In *Proceedings of the 1st International Semantic Web Conference (ISWC 2002)*, 2002.

[MW08] Gary Marchionini and Ryen White. Find What You Need, Understand What You Find. *International Journal of Human Computer Interaction*, 23:205–237, 2008.

[MWB+11] R. Mahendra, L. Wanzare, R. Bernardi, A. Lavelli, and B Magnini. Relational Patterns from Wikipedia: A Case Study. In *Proceedings of the 5th Language and Technology Conference*, 2011.

[NGPC12] Andrea Giovanni Nuzzolese, Aldo Gangemi, Valentina Presutti, and Paolo Ciancarini. Type inference through the analysis of Wikipedia links. In *Proceedings of Linked Data on the Web Workshop (LDOW2012)*, 2012.

[Nie93] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., 1993.

[Nie94] Jakob Nielsen. Estimating the number of subjects needed for a thinking aloud test. *International Journal of Human-Computer Studies*, 41:385 – 397, 1994.

[NLH07] Roberto Navigli, Kenneth C. Litkowski, and Orin Hargraves. SemEval-2007 Task 07: Coarse-Grained English All-Words Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, 2007.

[NP12] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[NPSR99] N Navarro-Prieto, M Scaife, and Y Rogers. Cognitive strategies in web searching. *Proceedings of the 5th Conference on Human Factors and the Web*, 2004:1–13, 1999.

[Ove97] Paul Over. TREC-6 Interactive Report. In Ellen M. Voorheer and Donna Harman, editors, *NIST Special Publication 500-240: The Sixth Text REtrieval Conference (TREC 6)*, pages 73–81. NIST, 1997.

[Ove01] Paul Over. The TREC interactive track: an annotated bibliography. *Information Processing & Management*, 37:369 – 381, 2001.

[OZG+11] Tope Omitola, Landong Zuo, Christopher Gutteridge, Ian C. Millard, Hugh Glaser, Nicholas Gibbins, and Nigel Shadbolt. Tracing the provenance of linked data using voiD. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*, 2011.

[Pag00] Stewart Page. Community Research: The Lost Art of Unobtrusive Methods. *Journal of Applied Social Psychology*, 30:2126–2136, 2000.

[PAG09] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. Semantics and complexity of SPARQL. *ACM Transactions on Database Systems*, 34:16:1–16:45, 2009.

[Par00] Soyeon Park. Usability, user preferences, effectiveness, and user behaviors when searching individual and integrated full-text databases: implications for digital libraries. *Journal of the American Society for Information Science*, 51:456–468, 2000.

[PB01] Carol Peters and Martin Braschler. Cross-language system evaluation: The CLEF campaigns. *Journal of the American Society for Information Science and Technology*, 52(12):1067–1072, 2001.

[PBMW99] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[Pet93] T.A. Peters. The history and development of transaction log analysis. *Library Hi Tech*, 1993.

[Pet08] Daniela Petrelli. On the role of user-centred evaluation in the advancement of interactive information retrieval. *Information Processing & Management*, 44:22 – 38, 2008.

[PHHD10] Axel Polleres, Aidan Hogan, Andreas Harth, and Stefan Decker. Can we ever catch up with the Web? *Semantic web : interoperability, usability, applicability*, 1:45–52, 2010.

[Phi00] Lawrence Philips. The Double Metaphone search algorithm. *C/C++ Users Journal*, 18:38–43, 2000.

[PKA10] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and building ontologies of linked data. In *Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, 2010.

[PL01] C. Perfetti and L. Landesman. Eight is not enough. http://www.uie.com/articles/eight_is_not_enough, 2001. Retrieved: August 2012.

[PMZ10] Jeffrey Pound, Peter Mika, and Hugo Zaragoza. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web (WWW 2010)*, 2010.

[PN10] Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich Word Sense Disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[PRS02] J. Preece, Y. Rogers, and H. Sharp. *Interaction design: beyond human-computer interaction*. J. Wiley & Sons, 2002.

[PS08] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3c recommendation, W3C, 2008.

[QF93] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1993)*, 1993.

[QL08] Bastian Quilitz and Ulf Leser. Querying distributed RDF data sources with SPARQL. In *Proceedings of the 5th European semantic web conference on The semantic web (ESWC 2008)*, 2008.

[RB83] R.E. Rice and C.L Borgman. The use of computer monitored data in information science and communication research. *Journal of the American Society for Information Science*, 34:247–256, 1983.

[RB01] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.

[RC10] Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2010)*, 2010.

[RD90] Detlef Rhenius and Gerhard Deffner. Evaluation of Concurrent Thinking Aloud using Eye-tracking Data. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 34:1265–1269, 1990.

[RHB92] S. E. Robertson and M. M. Hancock-Beaulieu. On the evaluation of IR systems. *Information Processing & Management*, 28:457–466, 1992.

[RLME05] Monique Reichert, Serge Linckels, Christoph Meinel, and Thomas Engel. Students perception of a semantic search engine. In *Proceedings of the IADIS Cognition and Exploratory Learning in Digital Age*, 2005.

[RM83] Teresa L. Roberts and Thomas P. Moran. The evaluation of text editors: methodology and empirical results. *Communications of the ACM*, 26:265–283, 1983.

[Rob81] S. E. Robertson. *The methodology of information retrieval experiment*, pages 9–31. Butterworths, 1981.

[Rob08] Stephen Robertson. On the history of evaluation in IR. *Journal of Information Science*, 34:439–456, 2008.

[RS67] A.M. Rees and D.G. Schultz. *A Field Experimental Approach to the Study of Relevance Assessments in Relation to Document Searching. Final Report to the National Science Foundation.* Case Western Reserve University, 1967.

[RSA04] Cristiano Rocha, Daniel Schwabe, and Marcus Poggi Aragao. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web (WWW 2004)*, 2004.

[RT11] Giuseppe Rizzo and Raphael Troncy. NERD : a Framework for Evaluating Named Entity Recognition Tools in the Web of Data. In *Proceedings of the 10th International Semantic Web Conference (ISWC 2011)*, 2011.

[Sak06] Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006)*, 2006.

[Sak07] Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information Processing & Management*, 43:531–548, 2007.

[San10] Mark Sanderson. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*, 4:247–375, 2010.

[Sar95] Tefko Saracevic. Evaluation of evaluation in information retrieval. In *Proceedings of SIGIR*, 1995.

[Sar07] Tefko Saracevic. Relevance: A review of the literature and a framework for thinking on the notion in information science. Part III: Behavior and effects of relevance. *Journal of the American Society for Information Science and Technology*, 58:2126–2144, 2007.

[SC90] A.L. Strauss and J.M. Corbin. *Basics of qualitative research: grounded theory procedures and techniques*. Sage Publications, 1990.

[Sch90] Y. Schabes. *Mathematical and Computational Aspects of Lexicalized Grammars*. Technical report. University of Pennsylvania, School of Engineering and Applied Science, Department of Computer and Information Science, 1990.

[Sch94] L Schamber. Relevance and information behavior. *Annual Review of Information Science and Technology (ARIST)*, 29:3–48, 1994.

[SDE+13] 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.

[SEN90] Linda Schamber, Michael B. Eisenberg, and Michael S. Nilan. A re-examination of relevance: toward a dynamic, situational definition. *Information Processing & Management*, 26:755–776, 1990.

[SFGM93] Michael Stonebraker, Jim Frew, Kenn Gardels, and Jeff Meredith. The SEQUOIA 2000 storage benchmark. *ACM SIGMOD Record*, 22:2–11, 1993.

[SGB98] Amanda Spink, Howard Greisdorf, and Judy Bateman. From highly relevant to not relevant: examining different regions of relevance. *Information Processing & Management*, 34:599–621, 1998.

[Sha86] Brian Shackel. Ergonomics in design for usability. In *People & computers: Designing for usability. Proceedings of the second conference of the BCS HCI specialist group*. Cambridge University Press., 1986.

[SHLP08] Michael Schmidt, Thomas Hornung, Georg Lausen, and Christoph Pinkel. SP2Bench: A SPARQL Performance Benchmark. *CoRR*, 2008.

[Shn86] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley Longman Publishing Co., Inc., 1986.

[SJ00] Karen Sparck Jones. Further reflections on TREC. *Information Processing & Management*, 36:37– 85, 2000.

[SJVR76] Karen Sparck Jones and C J K Van Rijsbergen. Information Retrieval Test Collections. *Journal of Documentation*, 32:59–75, 1976.

[SKCT87] T. Saracevic, P. Kantor, A. Chamis, and D. Trivison. Experiments on the Cognitive Aspects of Information Seeking and Information Retrieving. Final Report and Appendices. Research report, National Science Foundation, Div. of Information Science and Technology, 1987.

[SKCT88] Tefko Saracevic, Paul Kantor, Alice Y. Chamis, and Donna Trivison. A study of information seeking and retrieving. I. Background and methodology. *Journal of the American Society for Information Science*, 39:161–176, 1988.

[SMDV06] Jonathan Sillito, Gail C. Murphy, and Kris De Volder. Questions programmers ask during software evolution tasks. In *Proceedings of the 14th ACM SIGSOFT international symposium on Foundations of software engineering*, 2006.

[SMHM99] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

[SMV08] Jonathan Sillito, Gail C. Murphy, and Kris De Volder. Asking and Answering Questions during a Programming Change Task. *IEEE Transactions on Software Engineering*, 34:434–451, 2008.

[Soe94] Dagobert Soergel. Indexing and retrieval performance: the logical evidence. *Journal of the American Society for Information Science*, 45:589–599, 1994.

[Spi02] Amanda Spink. A user-centered approach to evaluating human interaction with Web search engines: an exploratory study. *Information Processing & Management*, 38:401–426, 2002.

[Spy92] J.H. Spyridakis. Conducting Research in Technical Communication: The Application of True Experimental Designs. *Technical Communication*, 39:607–624, 1992.

[Su92] Louise T. Su. Evaluation measures for interactive information retrieval. *Information Processing & Management*, 28:503 – 516, 1992.

[Su98] Louise T Su. Value of search results as a whole as the best single measure of information retrieval performance. *Information Processing & Management*, 34:557 – 579, 1998.

[Su03] Louise T. Su. A comprehensive and systematic model of user evaluation of Web search engines: II. An evaluation by undergraduates. *Journal of the American Society for Information Science and Technology*, 54:1193–1223, 2003.

[SZ05] Mark Sanderson and Justin Zobel. Information retrieval system evaluation: effort, sensitivity, and reliability. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR2005)*, 2005.

[TA10] Thomas Tullis and William Albert. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Morgan Kaufmann, 2010.

[Tag97] R. Tagliacozzo. Estimating the satisfaction of information users. *Bulletin of the Medical Library Association*, 65:243–249, 1997.

[Tau55] M. Taube. *Cost as the Measure of Efficiency of Storage and Retrieval Systems*. Defense Technical Information Center, 1955.

[Tau65] M. Taube. A note on the pseudo-mathematics of relevance. *American Documentation*, 16:69–72, 1965.

[TCA77] J. Tessier, W.W. Crouch, and P Atherton. New Measures of User Satisfaction With Computer Based Literature Searches. *Special Libraries*, 1977.

[TCC+10] Giovanni Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and Stefan Decker. Sig.ma: live views on the web of data. In *Proceedings of the 19th international conference on World Wide Web (WWW2010)*, 2010.

[TCRS07] Thanh Tran, Philipp Cimiano, Sebastian Rudolph, and Rudi Studer. Ontology-based interpretation of keywords for semantic search. In *Proceedings of the 6th international and 2nd Asian semantic web conference (ISWC2007+ASWC2007)*, 2007.

[TH01] Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, 2001.

[TH06] Paul Thomas and David Hawking. Evaluation by comparing result sets in context. In *Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM 2006)*, 2006.

[TKB10] Jonas Tappolet, Christoph Kiefer, and Abraham Bernstein. Semantic Web Enabled Software Analysis. *Journal of Web Semantics*, 8:225–240, 2010.

[TKP04] Giannis Tsakonas, Sarantos Kapidakis, and Christos Papatheodorou. Evaluation of user interaction in digital libraries. In *Notes of the DELOS WP7 Workshop on the Evaluation of Digital Libraries*, 2004.

[TLN06] C. W. Turner, J. R. Lewis, and J. Nielsen. Determining usability test sample size. *International Encyclopedia of Ergonomics and Human Factors*, pages 3084–3088, 2006.

[TM01] Lappoon R. Tang and Raymond J. Mooney. Using Multiple Clause Constructors in Inductive Logic Programming for Semantic Parsing. In *Proceedings of the 12th European Conference on Machine Learning*, 2001.

[TOD07] Giovanni Tummarello, Eyal Oren, and Renaud Delbru. Sindice.com: Weaving the Open Linked Data. In *Proceedings of the 6th International and 2nd Asian Semantic Web Conference (ISWC/ASWC2007)*, 2007.

[TS89] Jean Tague and Ryan Schultz. Evaluation of the user interface in an information retrieval system: A model. *Information Processing & Management*, 25:377–389, 1989.

[Ts92] Jean Tague-sutcliffe. The pragmatics of information retrieval experimentation, revisited. *Information Processing & Management*, 28:467–490, 1992.

[TS05] Diana Tabatabai and Bruce M. Shore. How experts and novices search the Web. *Library & Information Science Research*, 27:222 – 248, 2005.

[TS07] J. A. Thom and F. Scholer. A comparison of evaluation measures given how users perform on search tasks. In *Proceedings of the 12th Australasian Document Computing Symposium*, 2007.

[TsB94] Jean Tague-sutcliffe and James Blustein. A statistical analysis of the TREC-3 data. In *Overview of the Third Text REtrieval Conference (TREC-3)*, 1994.

[TSV99] Rong Tang, William M. Shaw, and Jack L. Vevea. Towards the identification of the optimal number of relevance categories. *Journal of the American Society for Information Science*, 50:254–264, 1999.

[UBL+12] Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over RDF data. In *Proceedings of the 21st international conference on World Wide Web (WWW 2012)*, 2012.

[UC11] Christina Unger and Philipp Cimiano. Pythia: compositional meaning construction for ontology-based question answering on the semantic web. In *Proceedings of the 16th international conference on Natural language processing and information systems (NLDB 2011)*, 2011.

[UCL+12] Christina Unger, Philipp Cimiano, Vanessa López, Enrico Motta, Paul Buitelaar, and Richard Cyganiak, editors. *Proceedings of Interacting with Linked Data (ILD 2012), at ESWC 2012*, 2012.

[UCLM11] C Unger, P Cimiano, V López, and E Motta. Proceedings of the 1st Workshop on Question Answering Over Linked Data (QALD-1), 2011.

[ULL+07] Victoria Uren, Yuangui Lei, Vanessa López, Haiming Liu, Enrico Motta, and Marina Giordanino. The usability of semantic search tools: a review. *The Knowledge Engineering Review*, 22:361–377, 2007.

[VB02] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2002)*, 2002.

[VH99] Ellen M. Voorhees and Donna Harman. Overview of the Eighth Text REtrieval Conference (TREC-8). In *TREC*, 1999.

[VH00] Ellen M. Voorhees and Donna Harman. Overview of the Ninth Text REtrieval Conference (TREC-9). In *TREC*, 2000.

[VH05] Ellen M. Voorhees and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval (Digital Libraries and Electronic Publishing)*. The MIT Press, 2005.

[Vic59a] B. C. Vickery. Subject analysis for information retrieval. In *Proceedings of the International Conference on Scientific Information*, 1959.

[Vic59b] B. C. Vickery. The structure of information retrieval systems. In *Proceedings of the International Conference on Scientific Information*, 1959.

[Vir90] R. A. Virzi. Streamlining the Design Process: Running Fewer Subjects. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 1990.

[Vir92] Robert A. Virzi. Refining the test phase of usability evaluation: how many subjects is enough? *Human Factors*, 34:457–468, 1992.

[VLL04] F. Vasilescu, P. Langlais, and G. Lapalme. Evaluating Variants of the Lesk Approach for Disambiguating Words. In *Proceedings of Language Resources and Evaluation (LREC 2004)*, 2004.

[Voo93] Ellen M. Voorhees. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1993)*, 1993.

[Voo98] Ellen M. Voorhees. Variations in relevance judgments and the measurement of retrieval effectiveness. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–323. ACM, 1998.

[Voo99] Ellen M. Voorhees. The TREC-8 Question Answering Track Report. In *Proceedings of TREC-8*, 1999.

[Voo01] Ellen Voorhees. Evaluation by Highly Relevant Documents. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[Voo02] Ellen Voorhees. The Philosophy of Information Retrieval Evaluation. In *Proceedings of the 2nd Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems (CLEF2001)*, 2002.

[Voo03] Ellen M. Voorhees. Overview of TREC 2003. In *TREC*, 2003.

[Voo09] Ellen M. Voorhees. Topic set size redux. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09. ACM, 2009.

[WBC07] Ryen W. White, Mikhail Bilenko, and Silviu Cucerzan. Studying the use of popular destinations to enhance web search interaction. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2007)*, 2007.

[WCSS00] E. J. Webb, D. T. Campbell, R. D. Schwartz, and L. Sechrest. *Unobtrusive measures*. Sage Publications, 2000.

[WER+10] S. N. Wrigley, K. Elbedweihy, D. Reinhard, A. Bernstein, and F. Ciravegna. Results of the first evaluation of semantic search tools. Technical report, SEALS Consortium, November 2010.

[WGCT11] Stuart N. Wrigley, Raúl García-Castro, and Cássia Trojahn. Infrastructure and workflow for the formal evaluation of semantic search technologies. In *Proceedings of the workshop on Data infrastructures for supporting information retrieval evaluation*, 2011.

[WHB91] S. Walker and M. Hancock-Beaulieu. *Okapi at City: An Evaluation Facility for Interactive IR.* Centre for Interactive Systems Research, City University, 1991.

[Wid95] Jennifer Widom. Research Problems in Data Warehousing. In *Proceedings of International Conference on Information and Knowledge Management*, 1995.

[Wil78] P. Wilson. *Two Kinds of Power: An Essay on Bibliographical Control.* University of California Press, 1978.

[Win90] William E. Winkler. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In *Proceedings of the Section on Survey Research*, 1990.

[Win99] William E Winkler. The state of record linkage and current research problems. In *Statistical Research Division, US Census Bureau.* Citeseer, 1999.

[WJD90] S. Walker, R.M. Jones, and R. DeVere. *Improving Subject Retrieval in Online Catalogues: Relevance, feedback and query expansion.* British Library Research Paper. British Library Board, 1990.

[WJLW85] John Whiteside, Sandra Jones, Paula S. Levy, and Dennis Wixon. User performance with command, menu, and iconic interfaces. *SIGCHI Bull.*, 1985.

[WKG+12] S. N. Wrigley, K.Elbedweihy, A.L. Gentile, V. Lanfranchi, and A.-S Dadzie. Results of the second evaluation of semantic search tools. Technical Report D13.6, SEALS Consortium, 2012.

[WRE+10] Stuart N. Wrigley, Dorothee Reinhard, Khadija Elbedweihy, Abraham Bernstein, and Fabio Ciravegna. Methodology and campaign design for the evaluation of semantic search tools. In *Proceedings of the 3rd International Semantic Search Workshop (SemSearch 2010)*, 2010.

[WUCB12] Sebastian Walter, Christina Unger, Philipp Cimiano, and Daniel Bär. Evaluation of a layered approach to question answering over linked data. In *Proceedings of the 11th international conference on The Semantic Web (ISWC 2012)*, 2012.

[WXZY07] Chong Wang, Miao Xiong, Qi Zhou, and Yong Yu. PANTO - A Portable Natural Language Interface to Ontologies. In *Proceedings of the 4th European Semantic Web Conference (ESWC 2007)*, 2007.

[XC96] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1996)*, 1996.

[Xie03] Hong Xie. Supporting ease-of-use and user control: desired features and structure of web-based online IR systems. *Information Processing & Management*, 39:899–922, 2003.

[YA06] Emine Yilmaz and Javed A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, 2006.

[ZK10] Junte Zhang and Jaap Kamps. A search log-based approach to evaluation. In *Proceedings of the 14th European conference on Research and advanced technology for digital libraries (ECDL 2010)*, 2010.

[Zob98] Justin Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 1998)*, 1998.

[ZWX+07] Qi Zhou, Chong Wang, Miao Xiong, Haofen Wang, and Yong Yu. SPARK: Adapting Keyword Query to Semantic Search. In *Proceedings of the 6th International and 2nd Asian Semantic Web Conference (ISWC2007+ASWC2007)*, 2007.

[ZZXY08] K. Zhou, H. Zha, G.-R. Xue, and Y Yu. Learning the Gain Values and Discount Factors of DCG. In *Proceedings of Beyond Binary Relevance Workshop, SIGIR 2008*, 2008.