UNIVERSITY OF SOUTHAMPTON

Faculty of Physical and Applied Sciences

Electronics and Computer Science

A BAYESIAN NETWORK MODEL FOR ENTITY-ORIENTED SEMANTIC WEB SEARCH

by

Christos L. Koumenides

Thesis for the degree of Doctor of Philosophy

November 2013

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL AND APPLIED SCIENCES
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE
WEB AND INTERNET SCIENCE RESEARCH GROUP

Doctor of Philosophy

A BAYESIAN NETWORK MODEL FOR ENTITY-ORIENTED SEMANTIC WEB SEARCH

by Christos L. Koumenides

The rise of standards for semi-structured machine processable information and the increasing awareness of the potentials of a semantic Web are leading the way towards a more meaningful Web of data. Questions regarding location and retrieval of relevant data remain fundamental in achieving a good integration of disparate resources and the effective delivery of data items to the needs of particular applications and users. We consider the basis of such a framework as an Information Retrieval system that can cope with semi-structured data.

This thesis examines the development of an Information Retrieval model to support text-based search over formal Semantic Web knowledge bases. Our semantic search model adapts Bayesian Networks as a unifying modelling framework to represent, and make explicit in the retrieval process, the presence of multiple relations that potentially link semantic resources together or with primitive data values, as it is customary with Semantic Web data. We achieve this by developing a generative model that is capable to express Semantic Web data and expose their structure to statistical scrutiny and generation of inference procedures. We employ a variety of techniques to bring together a unified ranking strategy with a sound mathematical foundation and potential for further extensions and modifications. Part of our goal in designing this model has been to enable reasoning with more complex or expressive information requests, with semantics specified explicitly by users or incorporated via more implicit bindings. The ground foundations of the model offer a rich and extensible setting to satisfy an interesting set of queries and incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar.

Empirical evaluation of the model is carried out using conventional Recall/Precision effectiveness metrics to demonstrate its performance over a collection of RDF-transposed government catalogue records. Statistical significance tests are employed to compare different implementations of the model over different query sets of relative complexity.

Contents

List of Figures

List of Tables

Declaration of Authorship

I, *Christos Koumenides*, declare that the thesis entitled

A Bayesian Network Model for Entity-Oriented Semantic Web Search

and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:

Koumenides, C., and Shadbolt, N. (2013) "Ranking Methods for Entity-Oriented Semantic Web Search". *Journal of the American Society for Information Science and Technology (JASIST).* Wiley-Blackwell (Accepted. In Press).

Koumenides, C., and Shadbolt, N. (2012) "Combining link and content-based information in a Bayesian inference model for entity search". In *Proceedings of the 1st Joint International Workshop on Entity-Oriented and Semantic Search (JIWES '12)*, SIGIR '12, ACM, 12-16 Aug, 2012, New York, NY, USA.

Koumenides, C., Salvadores, M., Alani, H., and Shadbolt, N. (2010) "Global Integration of Public Sector Information". In *Web Science Conference 2010 (WebSci '10)*, 26-27 April, 2010, Raleigh, NC, USA.

Signed: *Christos Koumenides*

Date: 13/11/2013

Acknowledgments

The research presented in this thesis was supported by the EnAKTing project funded by the Engineering and Physical Sciences Research Council under contract EP/G008493/1. Many thanks to my supervisors Prof. Nigel Shadbolt and Manuel Salvadores for their invaluable input and guidance on several aspects of my thesis, Harith Alani for his guidance during the early stages of my studies, my colleagues at EnAKTing and particularly Tope Omitola, Susan Davies, Ian Millard, Gianluca Correndo and Igor Popov, Lance Draper for his technical support, my girlfriend Eleni Kotsapa and all my friends and family for their support during the last few years.

Chapter 1

Introduction

This thesis examines the development of an Information Retrieval (IR) model to support text-based search over formal Semantic Web (Berners-Lee et al., 2001; Shadbolt et al., 2006) knowledge bases. We present and evaluate a model that adapts Bayesian Networks (Pearl, 1988) as a unifying modelling framework to provide ranking support to entity search in semantic data graphs. We make an effort to situate our work within the broader area of *semantic search* and offer extensive coverage of other modelling and algorithmic solutions that are of similar nature to our work.

Specific contributions of the thesis include:

- A formal model for entity-oriented search over semi-structured (graph-based) data, and in particular, Semantic Web data. The model provides a reasoning[1] basis that can cope with the inclusion of semantic relations in queries.
- A novel application of Bayesian Network theory. The thesis seeks to contribute to a better understanding of the use of Bayesian inference networks to support entity search in semantic knowledge bases.
- A cost-effective methodology for evaluating Semantic Web search models and a reference collection ready for prototyping and training of methods.

[1]The term "reasoning" here has a more general meaning than its figurative use in Semantic Web research (as in "DL Reasoning"). We will occasionally use the term in the thesis to refer to probabilistic and statistical reasoning, such as Bayesian inference.

- An in-depth review and classification of ranking methods for entity-oriented Semantic Web search.
- Recommendations on ways to organise and prioritise future research in semantic search.

In the following section, we introduce the area of semantic search from a broad point of view and outline the mainstream research directions in the field. Subsequently, Section 1.2 presents the scope of the thesis, in perspective to the more general area of entity-oriented search. Section 1.3 presents our motivation with the support of a detailed coverage of an example use case. Section 1.4 presents the outline of the remaining thesis.

1.1 Semantic Search

Semantic search (Baeza-Yates et al., 2008) is considered by many as the natural evolution of current search technology. While many conventional retrieval models have been proven to work effectively and efficiently over coarse document collections, there are many inherent obstacles to overcome when focus starts to shift towards items of finer granularity. Arguably, current search technologies are hindered by their limited understanding of user queries and ability to reason with more complex information requests requiring restrictions and finer specifications at the level of objects. Product search is a typically cited example of this realm. Traditional approaches to IR often treat documents as collections or bags of individual words, and their correspondence to a similar representation of user queries generally determines their level of similarity. This notion has often been coupled with simple forms of Natural Language Processing (Baeza-Yates, 2004) and features based on links, such as popularity and usage when search is conducted over Web-accessible documents. More elaborated retrieval models have also evolved in an effort to include information related to the classification of content inside documents, such that to prioritise selections based on where query terms

are found within the documents (whether part of a title, body, anchor text, etc.). The idea of semantic search is to diverge from this coarse view and sometimes monotonic treatment of documents to a more finer perspective, one that is able to exploit and reason intelligently with granular data items, such as people, products, organisations or locations, whether that is to complement document retrieval or facilitate different forms of search.

The advent of the Semantic Web (Berners-Lee et al., 2001; Shadbolt et al., 2006) is seen as an appealing vision for achieving deeper and better integration of data and information and, consequently, better understanding of the constructs and challenges for working in a semantic environment. Making searches semantic is about operating in an environment where symbols, documents and other resources are given well-defined meaning. The Semantic Web is about exposing structured information on the Web in a way that its semantics are grounded on well-defined and agreed-upon vocabularies. Through the established efforts of a number of online communities, there is now a large corpora of structured data in various formats (RDF[2], RDFa, XML[3], Microformats[4]) available for public consumption. Semantic Web repositories published as Linked Data[5] are estimated at a size of over 100 billion triples today. These include datasets pertaining to e-government, editorials, e-commerce, entertainment, scientific, encyclopaedic and possibly many other forms.

The availability of data on the Web has often served as an important vehicle for the development and investment into the next surge of Web and search technologies. Data integration and other compelling solutions are regularly explored for such tasks as data analysis, comparison, cataloguing, scheduling, etc. (Domingue et al., 2011). However, the usefulness of Web data is clearly dependent on the ease by which it can be discovered and consumed by others. In the context of search, there is a growing interest for solutions to alleviate access barriers and promote consumption of public data via ease

[2]W3C. Resource Description Framework (RDF): `http://www.w3.org/RDF/`
[3]Wikipedia. XML: `http://en.wikipedia.org/wiki/Xml`
[4]Wikipedia. Microformat: `http://en.wikipedia.org/wiki/Microformat`
[5]Linked Data: `http://linkeddata.org`

of discovery and reuse. This is evident in the efforts of various propositions to harvest, index and provide fast lookups over available data on the Web e.g. the Sindice[6] and SWSE[7] platforms providing flexible keyword-based retrieval over large volumes of data. At the same time, considerable research in the literature has focused on exploiting the availability of semantics to either enhance or complement document retrieval. This is also evident in the efforts of contemporary search engines to exploit data graphs in their search processes e.g. Google's Knowledge Graph[8]. Common to both cases is a shift of orientation from a Web of documents to a Web of objects, which raises new challenges to conventionally successful search processes and newly developed techniques.

1.1.1 Research Directions in Semantic Search

Semantic search is a dynamic area of research. The application area and realisation of different approaches has been very diverse and sometimes even lacking a common set of ideas. Information search utilising Semantic Web and other data graphs raises many challenging issues, including the modelling of queries and the definition of "documents" in response to queries. To these ends, there is widespread research covering developments across several distinct areas that do not necessarily coincide, although can rightfully be classified under the overall realm of semantic search. Some of the more established areas have received considerable attention and have been the focus of several academic conferences and workshops. We can identify mainstream research associated with a number of areas based on the orientation and focus of different developments, including:

- *Document-oriented search*, where the focus is on retrieval of documents, but using various ontological techniques to enhance document retrieval. For example, works

[6] Sindice. The semantic web index: `http://sindice.com`

[7] Semantic Web Search Engine (SWSE): `http://swse.org`

[8] Google Inside Search. The Knowledge Graph: `http://www.google.co.uk/insidesearch/features/search/knowledge.html`

that explore the combination of semantic metadata and other document features to improve retrieval performance or augment document lists with relevant data pulled from the Semantic Web (Vallet-Weadon et al., 2005; Guha et al., 2003; Fernandez et al., 2008; Han and Chen, 2006).

- *Multimedia search*, where formal representations of domain ontologies and semantic annotations are used for indexing and searching digital multimedia content, such as audios, images and movies (Linckels et al., 2007; Wei and Barnaghi, 2007; Celino et al., 2006; Ding et al., 2004a). Multimedia search may be thought of as a special case of entity search, except indexable features are usually the product of special processing peculiar to digital content, such as speech recognition, collaborative tagging or segment detection.

- *Association search*, where the focus is on discovery and interpretation of direct and indirect associations between resources (Sheth et al., 2004). The motivation here is that complex relationships can capture the meaning of resources and being able to extract the most obscured relations can provide essential insight information. Potential uses have been realised in a number of areas, including national security applications, such as being able to determine whether a flight passenger is known to be associated with an organisation on the watch list (Sheth et al., 2005).

- *Entity-oriented search*, where the focus is on retrieval of resources at the granularity of objects, such as products, people, organisations, etc. Entity search is a very active area of research, capturing developments that span a wide range of activities, from simple keyword and parameterised query algorithmic solutions to more elaborate design models for iterative and exploratory search (Uren et al., 2007; Hildebrand et al., 2007).

Entity search is a well-documented theme in the literature, lending itself to a wide perspective of research activities. The Semantic Web community has recently organised the Semantic Search Challenge[9], aiming to prioritise and evaluate research into "ad-hoc object retrieval" utilising Semantic Web graphs (Halpin et al., 2010; Pound et al., 2010). The outcome from the series has been a standard reference collection for conducting and evaluating experiments. Outside the mainstream Semantic Web research, the theme has appeared in a number of research tracks at the celebrated TREC[10] and INEX[11] conference series. The TREC Enterprise Track was initiated in 2005 with an expert search task (Balog et al., 2012) and the more recent TREC Entity and INEX Entity Ranking Tracks (Balog et al., 2010; Demartini et al., 2010) deal with searches at the entity level. These focus largely on entities represented as "pseudo documents" composed of virtual organisations of content from Wikipedia and other homepages. In the database community, keyword and natural language based search in databases is again a historic theme in the literature (Chen et al., 2009).

1.1.2 Contemporary production systems

Google's knowledge graph[12] is a reflective example of a popular search engine utilising semantic data graphs to enhance and complement its document search results. The idea of augmenting document lists with relevant semantic data has been investigated earlier by Guha et al. (2003), although Google materialised the concept into a real life production system. The knowledge graph is made of a large aggregation of data from a number of online sources (Freebase, Wikipedia), currently containing more than 500 million entities and 3.5 billion relations/properties[13]. The knowledge graph is a critical step from

[9] Semantic Search Challenge: `http://semsearch.yahoo.com`

[10] Text REtrieval Conference (TREC): `http://trec.nist.gov/`

[11] Initiative ofr the Evaluation of XML retrieval (INEX): `https://inex.mmci.uni-saarland.de/`

[12] Google Inside Search. The Knowledge Graph: `http://www.google.co.uk/insidesearch/features/search/knowledge.html`

[13] Google Official Blog. Introducing the Knowledge Graph: `http://googleblog.blogspot.co.uk/2012/05/introducing-knowledge-graph-things-not.html`

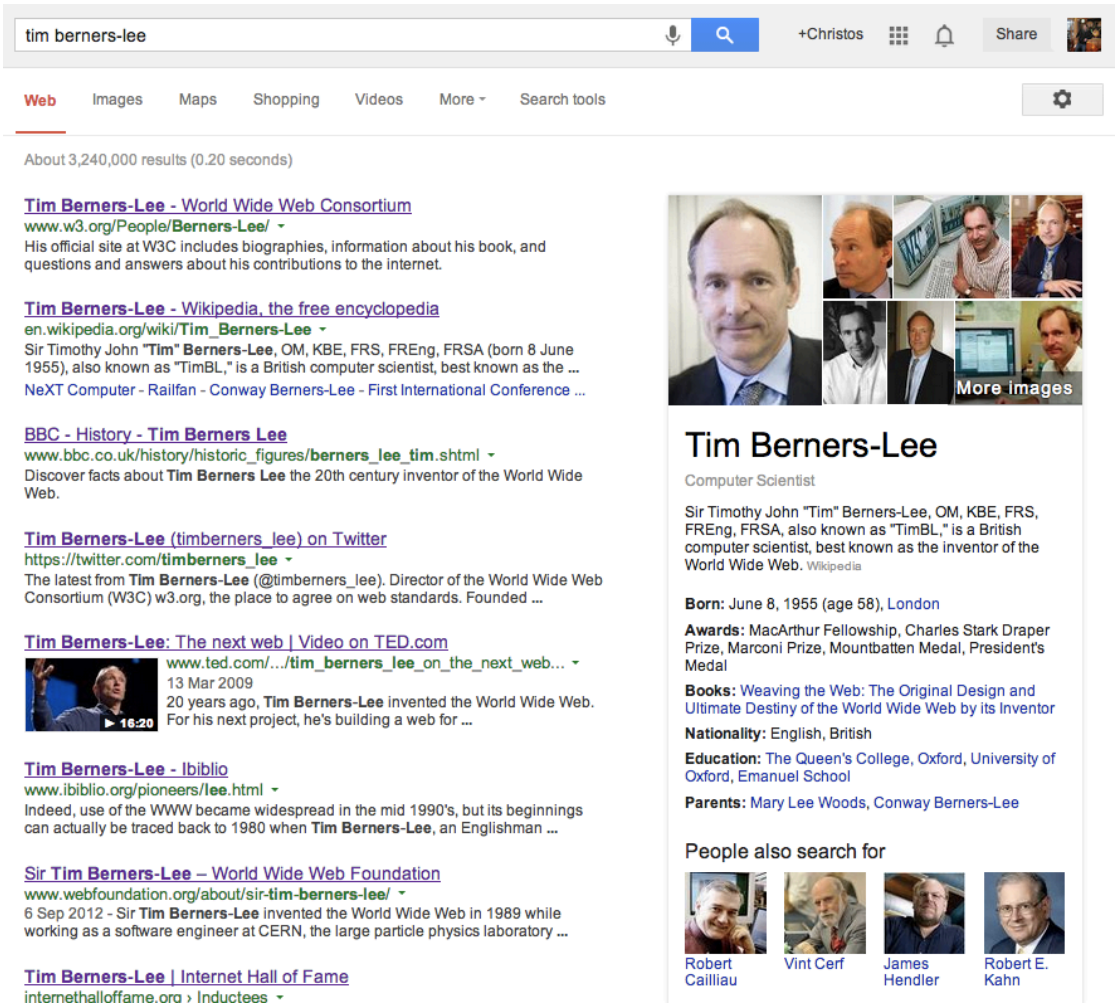


Figure 1.1: Google search for "Tim Berners-Lee".

a viable search engine company, making effective use of the collective intelligence of the Web to understand and reason with user-generated data.

Google utilises the knowledge graph to enhance search results by augmenting traditional lists of documents with what appears to be a precise match of entities. A brief investigation can indicate that this happens primarily when queries have a clear correspondence with the underlying data. The results retrieved from the knowledge graph involve deeper material than simple identifiers, including maps, statistics, weather reports and

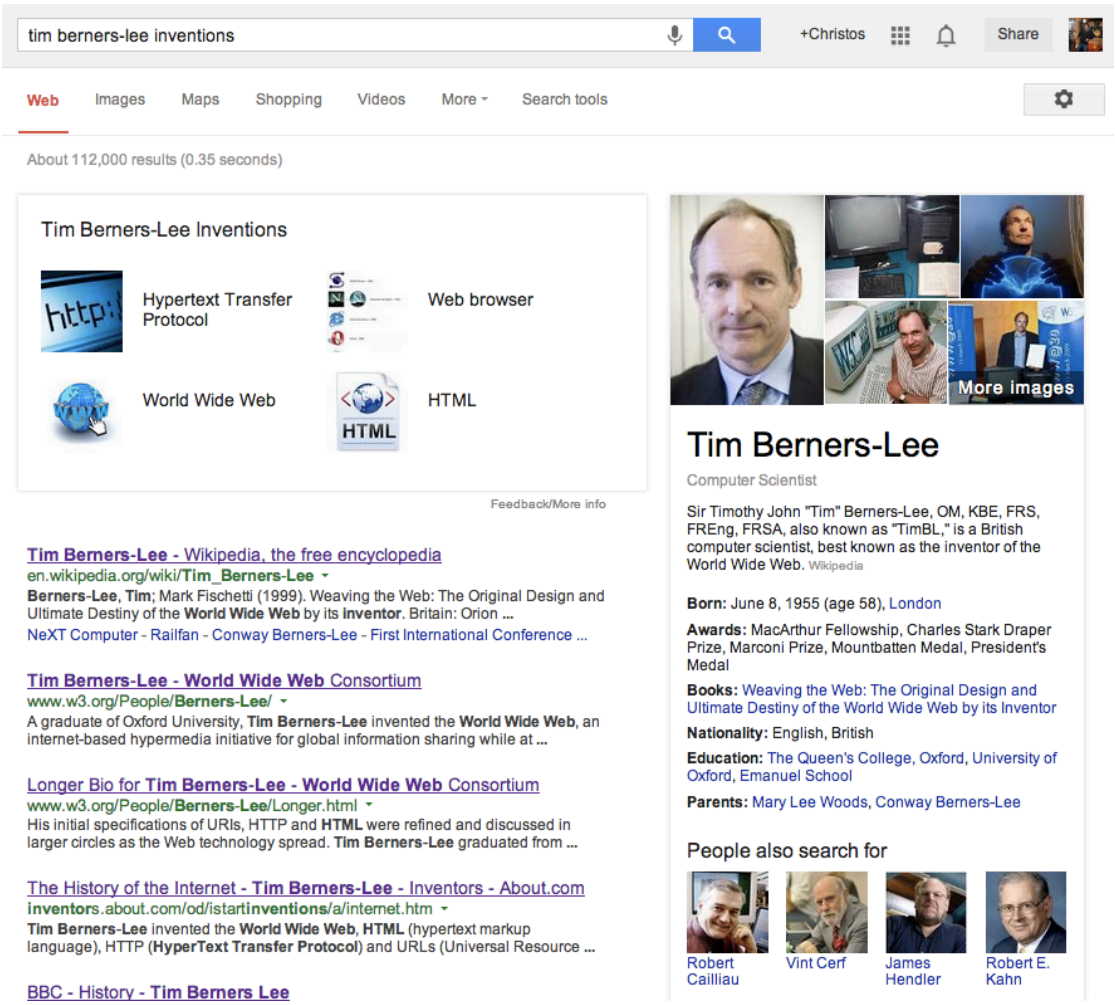


Figure 1.2: Google search for "Tim Berners-Lee inventions".

points of interest if places/locations are matched, or pictures, birthdays, education and work-related information if people are identified. One can also find detailed information on nutritional information for various foods and material. Accompanying the search results sometimes appears a list of similar search results, such as things that other people also search for. The system appears quite effective over more deterministic queries (such as names and properties of popular and well-defined "things"), but appears to take little risk when there is no on-to-one mapping between the query and the underlying store. In

the latter case, the engine resolves to the customary list of document identifiers and brief descriptive info. For example, we can resolve a query for "Tim Berners-Lee" (Figure 1.1), but will get **no** results from the knowledge graph for a query for "Tim Berners-Lee w3c" or "Tim Berners-Lee Web". Results are provided, however, for "Tim Berners-Lee www" and "Tim Berners-Lee inventions" (Figure 1.2), with the same person entity retrieved and displayed alongside the main results. The latter offers an additional section with various inventions associated with Tim Berners-Lee (Figure 1.2).

Wolfram Alpha[14] is another example of a contemporary system that utilises semantics in search. The platform focuses on computable knowledge, rather than document lists, and provides descriptive answers to user queries. This is unlike customary search engines that focus on lists of identifiers to either internal or external resources/documents. The platform is advertised as an "ambitious, long-term intellectual endeavor" intended to deliver increasingly sophisticated capabilities over completely free-form input[15]. In fact, it goes a long way beyond simple keyword processing. The system can provide detailed analyses over a wide variety of complex textual queries e.g. resolve complex mathematical equations (plot algebraic functions, solve derivatives and integration), physics, music compositions (e.g. give a keyboard and pentagram representation of musical notes e.g. $F\#G\#$), predict populations (queries such as "population of Cyprus in 2030" are possible), factual questions (e.g. "1980 Nobel Prize in Chemistry"), comparisons from available statistical datasets (e.g. "UK vs Germany education expenditures"), and many other interesting queries[16].

Wolfram Alpha is in relatively early production (launched in mid 2009). False positives[17] and generally approximated answers to queries can be found where the input proves ambiguous or the data is not available. Being a commercial system, there is little

[14] Wolfram Alpha: `http://www.wolframalpha.com`

[15] Wolfram Alpha, About: `http://www.wolframalpha.com/about.html`

[16] Wolfram Alpha, Examples by Topic: `http://www.wolframalpha.com/examples/`

[17] Wikipedia. Type I and type II errors: `http://en.wikipedia.org/wiki/Type_I_and_type_II_errors`. Also see Manning et al. (2008), Chapter 8.

mention of where the data comes from and how it is curated[18]. Internal knowledge bases are utilised, however, and these potentially involve extensive semantics and ontologies[19].

1.2 Scope and Overview of the Thesis

The scope of this thesis is to study the problem of entity search on the Semantic Web from the perspective of a new and expressive IR model capable to reason with accuracy over semi-structured information resources. Compared to documents on the Web which are commonly seen as flat sequences of words, semi-structured data are generally more complex and diverse items and can denote any kind of entity, whether a person, location, document, product, etc. Such data (whether in Linked Data or other formats) can have their own defined schemas that provide some kind of structure, which may or may not follow strict rules or that may change over time as the information in a given repository changes. There is therefore no fixed a priori schema that most Semantic Web or otherwise data resources are bound on, and this poses significant obstacles in terms of a semantic search model that can function in a generic, yet effective, manner.

For the above reasons, we envision a model that is, first of all, *entity centric*, in that it diverges from the concept of flat unstructured documents to the concept of objects as depicted generically on the Semantic Web. An entity description is a more complex data object, which, in its most fundamental sense, can be seen as a set of attribute and value pairs and relations with other entities i.e. in the form of a triple or directed labelled graph. On the Semantic Web, entities are generally treated as anything that is addressable by a URI and can serve as the subject of a description (where a description is more formally depicted by a collection of triples, which may serve as the concise representation of a resource)[20]. At the same time, the model needs to remain flexible to extensions, such

[18] Wolfram Alpha, Data FAQ: `http://www.wolframalpha.com/faqs5.html`
[19] Wolfram Alpha, API FAQ: `http://products.wolframalpha.com/api/faqs.html`
[20] We discuss the search context in more detail in Section 1.3.1 and Chapter 2.

that to maintain an evolving state and be adoptable to changes, and be independent of any predefined schema or a set of available semantic connections to be functional.

Our focus is on fully-automatic query processing, whereby information requests are given in free-form keyword or natural language queries. Part of our goal in designing the model has been to enable reasoning with more complex or expressive information requests, with semantics specified explicitly by users or incorporated via more implicit bindings. We adapt a reasoning basis that enables the inclusion of semantic relations in queries, although we do this in a way that does not restrict the model's utility and scope, but rather enhances its effectiveness when they are specified i.e. resolving queries of the form "find me all the movies directed by Francis Ford Coppola". This is intended to allow more experienced users or application needs to take advantage of more expressive query formulation to fine-tune their interaction with the system. As we will present throughout the thesis, such association semantics have a natural fit in the model and can act as either explicit provisions in the inference or as more implicit propagation impulses that can alter the impact of evidence as traversed through a set of probabilistic dependencies. Our scope remains on fully-automatic query processing, whether additional query semantics (mostly pertaining to associations) are specified or not. The ground foundations of the model offer a rich setting to satisfy an interesting set of queries.

The proposed model employs a variety of techniques to leverage the available semantics in the data (mostly focusing on interrelations between data items) to bring together a unified ranking procedure with a sound mathematical foundation and potential for further extensions and modifications. We achieve this by developing a *generative* (Bishop et al., 2006) Bayesian Network (BN) model that is capable to express the explicit semantics associated with resources and expose them to statistical scrutiny and inference procedures. The goal of our translation is to devise a generative model for projecting the directed labelled graph (DLG) manifestation of knowledge bases to a form of directed acyclic graph (DAG), on which we can delicate retrieval of resources to an evidential reasoning

process. The resulting model is not necessarily restricted to Semantic Web data, since a translation from a DLG model can have a broader perspective. Dependence implications from Semantic Web assertional and terminological constructs will be treated by the same general-purpose statistical schemes. As it is customary with BNs in IR, we treat the model as an expressive architectural framework on which we can approximate reasoning using various generic functions of standard IR schemata (e.g. functions to estimate term frequency, field weighting, and link proximity). The ground foundations of the model offer a rich setting to incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar.

Our study focuses on presenting the model from its initial inception, specification, instantiation on a particular data collection, and evaluation to demonstrate its indicative performance. The focus of the evaluation is on the quality of search results produced by the model i.e. how the model responds and what it delivers when applied over a realistic data collection. Since we are focusing on a model that retains a generic character, although remains parametric to fine-tune its focus and performance on particular types of queries, we carry out an evaluation procedure using standard IR effectiveness metrics that can generalise performance over a range of queries.

To summarise, in this thesis we investigate the problem of query-dependent ranking of entities from a knowledge base as responses to user queries. Furthermore, we focus on a single mode of interaction, where queries are provided in either free-form or semi-structured natural language queries. For the bulk of the remaining thesis, we suppose that the system provides an interface via which a query can be constructed and sent for processing.

1.2.1 What the thesis is not about

Information Retrieval is a broad area of research with various areas of concentration. There are a number of dimensions that are equally fundamental in solving or comple-

menting a solution to the problem, which we do not make an attempt to investigate. For example, we do not investigate if more advanced user interfaces, such as multi-facet views, or query refinement techniques, such as query suggestion and relevance feedback, can be effectively integrated with our model to support or enhance the utility of end-users. There are prospects for further research concerning our work, and user interface integration is certainly one of them.

Another topic that we do not cover is data acquisition and indexing, such as crawling and organisation of data into an inverted index structure for efficient search. These are core aspects concerning IR systems, particularly the process of indexing data. We support the belief, however, that effectiveness is the foremost criterion for an IR system. Once a technique is established as potentially useful, then focus can shift to finding efficient implementations utilising appropriate index structures, compression techniques and query pruning heuristics.

1.3 Motivation

The thesis is foremost motivated by the increasing availability of semi-structured data on the Web, which brings an interesting frontier for research into appropriate interaction mediums and search/retrieval schemes.

Ranked keyword search over graph-structured data has attracted much attention recently for a number of reasons. Keyword-based search tools generally do not require users to master a complex query language or understand the underlying data schema to be able to interact. In effect, they are a very attractive frontier for research into scalable semantic search engines that can cope with multiple heterogeneous data collections. Furthermore, ranked keyword search can generally function as the starting point for further exploration and search, and users have grown to be accustomed with this setting. Even complex systems based on articulated interfaces often require an initial starting

point for users to engage in further interaction. Keywords can be used to pinpoint objects of interest, after which a system can provide additional menus and filters to incrementally reduce the size of the results or construct more expressive queries. It remains vital, therefore, that an effective and flexible retrieval model is available for the core functionality of a system, whether that is to be treated as a stand-alone facility or part of a bigger complex of tools.

In the following section, we provide a detailed use case to motivate our work over a pragmatic scenario involving a collection of government catalogue records and the development of global Public Sector Information portal.

1.3.1 Example use case

In early 2010, we initiated the development of a Public Sector Information (PSI) catalogues aggregator service as part of the EPSRC EnAKTing project. A related publication is available (Koumenides et al., 2010) and an online platform with several of our experimental results published in various forms (data visualisations and other linked data browsing utilities)[21]. The project aimed to aggregate the online catalogues of various PSI portals and promote a set of end-user facilities for viewing and searching the contents of the catalogues. We managed to retrieve in raw HTML/RDFa and CSV formats the contents of four government portals (*Data.gov*, *Data.gov.uk*, *Data.australia.gov.au*, *Opsi.gov.uk*), summing up to over 9,000 records. Of the four portals, three of them were further selected to carry on with the experiments (*Opsi.gov.uk* came offline as obsolete and was therefore removed from the experiments).

The contents of the catalogues were cleansed and refurbished into RDF format, which resulted to three separate light-weight schemata. These are illustrated in Appendix B. Where possible, we made an effort to project clearly-defined entities (such as departments,

[21] PSI Catalogues Aggregator: `http://catalogues.psi.enakting.org` (there are two main sections on the site, named Federator and Analyzer, which link to the various applications available)

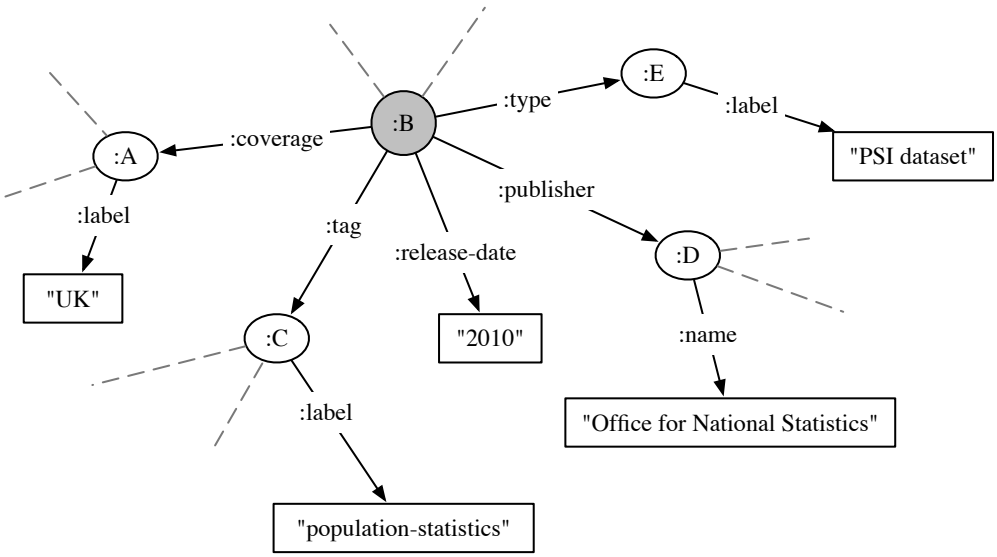


Figure 1.3: Fraction of a data sample depicting a Public Sector Information record instance.

tags, locations, jurisdictions, and licenses) onto first-class objects and assign them URIs to be dereferenceable on the Web. This resulted in a more general dataset made of multiple tightly-coupled sub-collections interrelated via the presence of object properties between them. As an example, consider the fraction of data in Figure 1.3, illustrating the correlation of a record, tag, location and publisher entities. Statistics about the catalogues are provided later in the thesis (during evaluation of the model).

Part of our goal in developing the PSI catalogues aggregator service was to surface the data under a single location for non-technical users to be able to search and visualise their contents. We made the following assumptions about the definition of users for the online portal:

1. Users would not be required to know anything about the format, schema, and any other classification information about the underlying data to be able to search and retrieve their contents.
2. Users would not be required to formulate programmatic queries, such as SPARQL[22]

[22] W3C. SPARQL 1.1 Query Language: `http://www.w3.org/TR/sparql11-query/`

or SQL[23], to retrieve contents from the catalogues.

These are sensible assumptions, since the portal was intended for public consumption and would not require its users to be familiar with Semantic Web or other expert technologies. The same criteria are assumed at the originating national portals of the catalogues. Furthermore, intricacies in the underlying data call for a more *flexible* retrieval model than one based on a purely logical query language (such as SPARQL). We will discuss these next.

At this point, the focus was not just on record entities, but we would like users to be able to retrieve departments, locations, and tags, which they could then follow through by clicking on associated entities to retrieve more information linked with them. We decided that free-form text search is the desirable option as a starting point for further exploration in our PSI aggregator service. **Our choice is based on the two factors mentioned above:** (1) preference criteria (assumptions outlined above), and (2) pragmatic restrictions from intricacies in the underlying data, which call for a more *flexible* retrieval model than one based on a purely logical query language. We discuss these points next.

1.3.1.1 Why free-form text search

Foremost, users are accustomed to such a setting, and with the proper modelling they would be able to get quick answers to arbitrary queries. We would like users to be able to type a natural language query, retrieve a set of entities and start clicking through them until they are satisfied with the results they obtain. This is purely a preference criteria, inline with our aforementioned assumptions about the users of the portal.

Another important factor arises upon inspection of the data, which we realised was not properly and uniformly classified across all four (now three) catalogues. For example, not all content was associated with metadata at the proper granularity, such as

[23] Wikipedia. SQL: `http://en.wikipedia.org/wiki/SQL`

Figure 1.4: Tag cloud of the most frequent tags in the UK catalog.

appropriate tags and other category metadata. These would be important for any sort of retrieval process, even if it means using a purely structural approach via a programmatic query language such as SPARQL. Some records would contain as many as 50 or more tags (on avg. there were 8 tags per record with approx. 2000 records lying above average - the highest being 368 tags in a single record), some of fine granularity, such as "bank-performance-and-conditions-ratios", while others would contain as few as 1 tag (approx. 10% of the records were linked to a single tag with a much broader granularity - e.g. "statistics"). There were a total $11,000$ unique tags in the dataset, while some records would contain no tags or any other category-related metadata. Worse yet, the four catalogues lacked a uniform vocabulary to organise the content into some global classification - for instance, tags associated with the catalogues contained multi-token, compound terms (such as "health-and-social-care" or "financial-and-contractual" or "marriages-cohabitations-civil-partnerships-and-divorces") with little cross-match and consistent use between the catalogues.

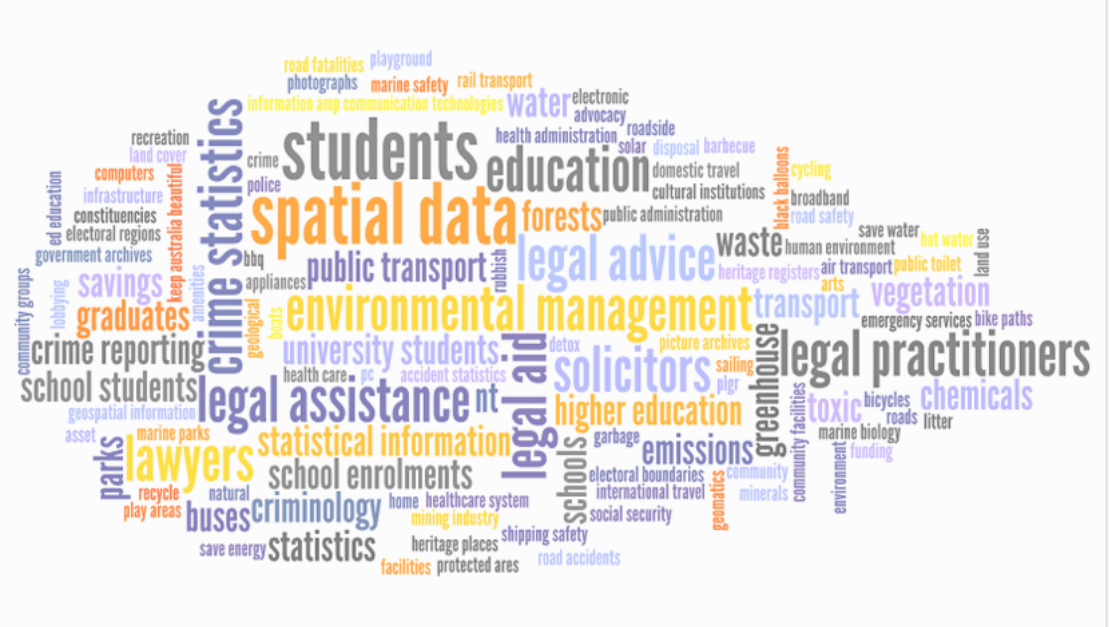


Figure 1.5: Tag cloud of the most frequent tags in the Australian catalog.

Observe the tag-cloud snapshots in Figures 1.4−1.7 for examples of the most frequent tags in the dataset. There is little or no overlap between the most heavily used tags, but when we look closer and find matches, most of them are only partial e.g. “health” and “health-well-being-and-care” in the UK catalogue with “health care” and “health administration” in the AU catalogue. These would need to be broken down to single terms if we were to find any potential mappings across the four catalogues.

From the outset, we knew that the situation would worsen as more data was inputed into the aggregator. We knew that a standard and global (worldwide) classification vocabulary was a long way from being enforced and implemented with consistency across all PSI data catalogues around the globe. The situation would only get worse as more catalogues were released in different languages (e.g. German, French, Greek, etc.), which would need to be translated, resulting in even more noise. More so, the situation could potentially get worse as we start to bring non government-released data to the portal.

A purely logical approach is able to retrieve resources that meet some logical condition. For example, SPARQL would be 100% effective/precise in retrieving all records tagged with the "population-statistics" tag. If we wanted, however, to resolve a query such as "population statistics", a purely logical approach would fall short, unless the underlying data is clearly classified and we knew beforehand what we want to search for (e.g. have a clear procedure for representing all potential information needs to SPARQL equivalents). Even if we knew what we wanted to search for though, what happens when there are records that do not contain the "population-statistics" tag? A developer using SPARQL could resolve to associating a *regular expression*[24] with the logical conditions of the query to increase coverage in the results i.e. retrieve all records that are associated to some literal value that contains the words "population" and/or "statistic" in some predefined order. One possible outcome of this would be to increase coverage by retrieving records that mention "population" or "statistics" in their description, title, or even associated department's title (e.g. "Office for National Statistics"). Most likely, some post-processing of results would need to occur in order to present them in some meaningful order.

The problem with the approach just described is that there is no sense of *ranking* in the results, unless post-processing of results occurs (which in essence forms a new retrieval model - one that combines the logical query language with some post-processing relevance model). The purely SPARQL approach would be mixing results without considering their *degree* of relevance to a query e.g. records tagged with "population-statistics", or published by a related statistics agency, or containing some order of the terms in the description or title, could all be positive answers. For a generic application that users can ask arbitrary queries, this would be a problem. We needed some form of relevance model.

[24] Wikipedia. Regular expression: `http://en.wikipedia.org/wiki/Regular_expression`

Figure 1.6: Tag cloud of the most frequent tags in the US catalog.

Figure 1.7: Tag cloud of the most frequent tags in the OPSI catalog.

1.3.1.2 Challenges ahead

We decided that free-form user input is the preferred method for interaction at the portal (at least as a starting point for further exploration and search). There are the following factors to take into account at this point:

1. How to resolve free-form user input over the data
2. How to present results to the user
3. How to engage the user in further browsing and exploration

As we envisioned the portal to encompass and provide access to several items of data, it is important to have a facility to search the catalogues dataset not only in its current state, but also as it would evolve into a larger integration of data. For example, we envisioned that jurisdictions, tags, and publishers in Figure 1.3 would in time be linked to other sources providing additional information and other entities associated with them. This would in essence lead to a more general silo of government data federated at the portal, such as dissemination methods and expenses, department locations, personnel, reports, etc. In other words, we wouldn't like to stop at the records in the search for data.

The model we present in this thesis aims to address the first fundamental question: how to resolve free-form user input over the data. Our use-case requires a model that can take any given dataset of a similar structure to the example given above (not just catalogue records) and resolve completely free-form input over it, with relative precision. Such a model would serve as a viable solution for establishing the entry point to our system. It would be desirable for the system to resolve queries such as "**datasets released in 2010**" or "**datasets published by Scottish Government**", but also less-refined queries, such as "**crime datasets**" or simply "**crime**". The model should behave similarly to other forms of entities, not just queries concerning datasets, particularly as more data would be added to the portal.

There are several important technical criteria to take into account when considering the aforementioned queries:

1. How would a search engine resolve queries when the sought-for entities are not directly associated with the terms in the query? Conventional IR systems (as implemented in common search engine libraries e.g. Apache Lucene[25] and Lemur[26]) work by establishing mappings between terms in the query and term indexes in the data (Manning et al., 2008). From these mappings, documents (resources in our case) are retrieved and processed further (ranked). Considering Figure 1.3, the problem arises with a query such as "population datasets", where the term "population" is only associated with the tag entity C. In other words, the query term "population" and the entity B (the correct answer to the query) are *indirectly* related via an intermediate entity C. In the same way, the term "dataset" is again only indirectly related to B via the entity E. It is also not unlikely for multiple intermediary nodes to exist between the matching terms and the potentially sought-for entities (a case that has not arisen in the catalogues dataset but not unlikely to exist). How such mappings are established by a search engine and whether they can be accomplished successfully via some form of graph exploration or propagation technique is the subject of research. Examples of graph exploration techniques from the literature are provided in the forthcoming chapter.
2. There is evidently a need to not only retrieve resources in response to, potentially ambiguous, keyword queries, but also to rank resources as more or less relevant to a given query. Conventional IR models treat queries as frames of evidence (for example, the terms "population" and "datasets" would be two pieces of evidence) to be mapped onto an underlying inverted index (Manning et al., 2008) (from a corpus of terms) to find associated resources that are somehow linked to those terms. The corpus acts as a general index to resources that are linked to it. This generic nature of search engines leads to a lack of a clear procedure to decide whether a resource from a dataset is a correct answer to a query. In other words,

[25] Apache Lucene: `http://lucene.apache.org`
[26] The Lemur Project: `http://www.lemurproject.org`

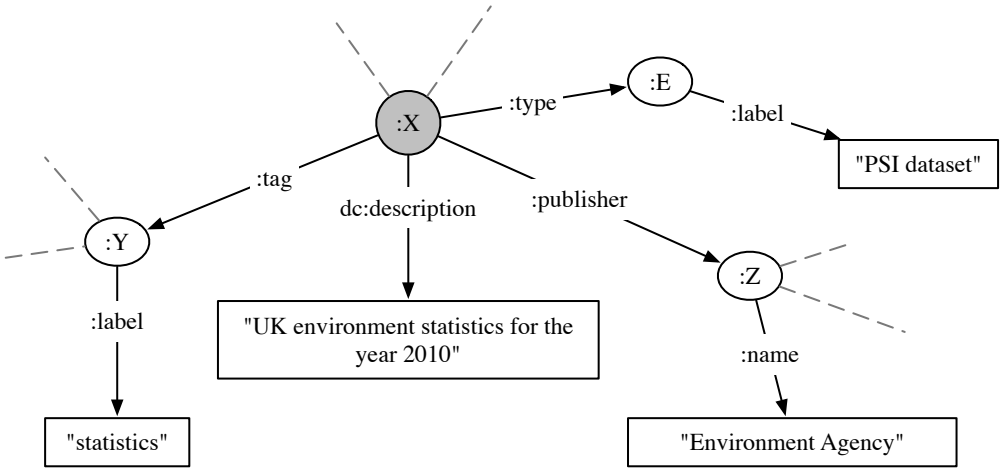


Figure 1.8: Fraction of a data sample depicting a record instance with a vague tag "statistics" classifying the record.

when resolving a query such as "2010 statistics" (consider Figure 1.3), there is no clear procedure for mapping these terms to "facts" in the knowledge base. When thinking in terms of a keyword search engine, over some arbitrary data, this is a key fact to take into account. Unless we expect an IR system to function over a predetermined set of queries and data (predetermined structure and content), then ranking will have a key role in the process.

Let us consider an example where the data is more vaguely classified with broader/-general information, as in Figure 1.8. The term "environment" from a query such as "environment statistics" only appears within a larger sentence in the description of the resource. There is no "environment" or "environment statistics" tag to give a precise classification for the resource. The term "environment" also appears in the name of the publisher of the resource. How would a search engine know that the publisher entity Z in not relevant to a search for "environment statistics" when there is no clear procedure to associate terms with the underlying data? An entity search model dealing with ambiguous queries (where there is an unclear correspondence with the underlying data) could resolve this by ranking the results

and aim to rank highest those entities that are somehow linked to most of the evidence (terms) in the query (just like a traditional search engine would attempt to do with documents (Manning et al., 2008)). In this case, an effective search engine would produce a result set where resource X is more relevant than resource Y, Z or E. The idea is the same to conventional document search engines, except the underlying data model is more complex in our case. The idea is to aim at as few false positives and negatives (Manning et al., 2008, Chapter 8).

3. Assuming that a retrieval model can somehow accommodate the above criteria, what would it take to resolve more complex natural language queries? For example, to what extend would queries of the form "UK datasets released in 2010" or "datasets published by UK Statistics Authority" be possible by the same retrieval model? Such queries contain evidence of object and datatype relations in them, hence have a more well-defined meaning. Ideally, we would like our keyword search engine to be able to differentiate and associate a higher relevance factor to datasets that were indeed published by the UK Statistics Authority, as opposed to datasets that were authored by the UK Statistics Authority. Note that *Data.gov.uk* contains statements about records that were both published and authored by different agents (as implied by the schema in Figure B.1), while the same agent may appear as both the author and publisher of a given dataset.

The above points help to motivate and outline the challenges ahead for our work on an information retrieval model to retrieve and rank entities from semantic data graphs in response to keyword queries. Keyword queries entail no predefined structure, therefore can be ambiguous to resolve. A search engine would require some form of statistical (or otherwise) reasoning basis to determine the proper paths to be traversed in the graph where correspondences with the query are established, or apply statistical methods to propagate and infer the impact of query evidence on the resources in the data. The various questions raised above are what we aim to address in this thesis via the proposed

model. The foremost criteria that we envision of such a model is to provide a facility for processing both simple and more complex keyword queries, and be flexible to data that does not pertain strictly to catalogue records.

1.4 Outline of the Thesis

Chapter 2 provides a technical review of semantic search methods used to support text-based search over formal Semantic Web knowledge bases. The focus is on ranking methods and auxiliary processes explored by existing semantic search systems, outlined within broad areas of classification.

Chapter 3 presents the topological properties of a Bayesian inference network model to support entity search in semantic knowledge bases. The chapter covers definition of system variables and dependence relations, query modelling, and example networks.

Chapter 4 unifies the model into a complete specification by presenting methods for quantifying conditional probabilities in the model, a ranking strategy, inference formulas for belief computation, and canonicalisation of complex interactions/dependencies.

Chapter 5 provides a detailed walkthrough of the model over a sample data graph. The example covers all aspects of the model discussed in Chapters 3 and 4 (translation from a directed labelled graph, observing a query, assignment of probabilities, and ranking via probabilistic inference).

Chapter 6 proceeds with instantiation of the model over a realistic data collection, training and configuration of system parameters, and evaluation of the model's effectiveness using standard Precision/Recall metrics. Statistical significance tests are employed to compare different implementations of the model over different query sets of relative complexity.

Chapter 7 revisits our objectives and main findings and discusses directions for future research.

Chapter 2

Ranking Methods for Entity-Oriented Semantic Web Search

2.1 Introduction

In this chapter, we provide a technical review of ranking methods used to support text-based search over formal Semantic Web knowledge bases. Our focus is on ranking methodologies and auxiliary processes explored by existing semantic search systems, with particular emphasis on methods that make use of the graph structure of Semantic Web data. Ranking models have been an integral part of Information Retrieval (IR) research and remain an active and challenging dimension in modern frameworks and data models. Throughout the review, we seek to obtain a deeper understanding of the architectural choices that play a role in supporting text-based search over Semantic Web data. For this reason, we focus on presenting a few topics in some detail. The presentation covers graph exploration and propagation methods, adaptations of classic probabilistic retrieval models, and query independent link-analysis via flexible extensions to the PageRank algorithm. The survey is not intended to be an exhaustive list of available architectures, but rather a detailed outline of reflective examples from the literature. Future research directions are discussed, including development of more cohesive retrieval models to unlock further potentials and uses, data indexing schemes,

integration with user interfaces, and building community consensus for more systematic evaluation and gradual development.

The forthcoming review maintains a strong Semantic Web orientation, as it is prevalent throughout the material selected for review. However, the techniques outlined are conceptually, and sometimes pragmatically, applicable to any type of data that pertains to a graph structure, particularly directed-labelled graphs, as will be outlined next. The presentation takes on a holistic view of developments in this area, both across the Semantic Web but also as supported by works in similarly related fields e.g. relational database[1] and XML[2] search. The selection of works has been driven by the availability of detailed and complete descriptions, and the need to capture a wide spectrum of techniques and architectural frameworks. The presentations follow a common outline: a detailed description of a characteristic operation is presented, followed by reflective examples of individual systems that explore or implement the operation in a given context. We give special emphasis on the evaluation procedures followed to demonstrate the performance of individual systems and any coupling involved with other methods to facilitate overall retrieval.

As a prelude to the main review, the following section offers a brief coverage of the data context and some of the fundamental challenges that naturally arise.

2.2 Search Context – Data and Challenges

2.2.1 The Semantic Web

The Semantic Web (Berners-Lee et al., 2001; Shadbolt et al., 2006) is an extension of the current Web that aims to underpin Web resources with machine-understandable data in order to optimise sharing, reuse and general handling of information. The infrastructure

[1] Wikipedia. Relational database: `http://en.wikipedia.org/wiki/Relational_database`
[2] Wikipedia. XML: `http://en.wikipedia.org/wiki/Xml`

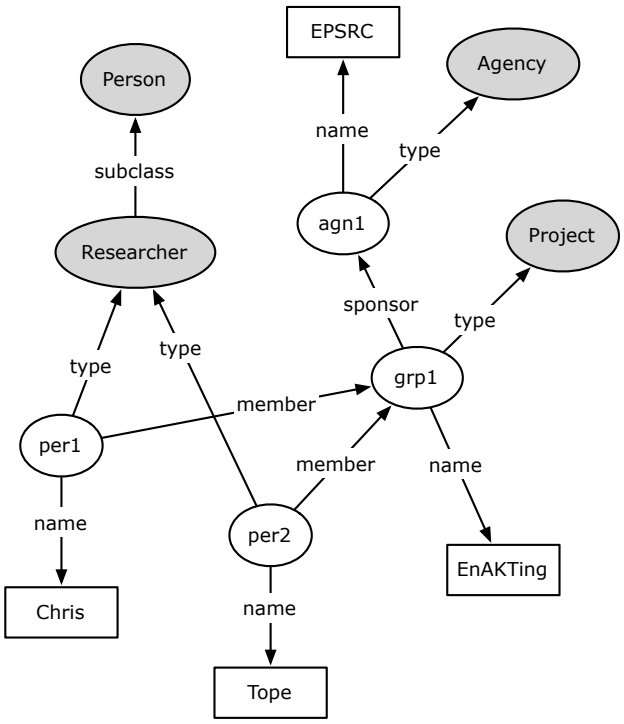


Figure 2.1: Example RDF data graph.

of the Semantic Web proposes a set of standardised technologies to effectively handle the global identification, modelling and querying of semi-structured data resources. The Resource Description Framework (RDF)[3] forms the foundation of data modelling languages on the Semantic Web and provides a syntax that enables the use of Uniform Resource Identifiers (URIs) to name resources. RDF is a flexible graph-based data model and provides a foundation for more advanced and expressive assertional languages.

Semantic Web data are maintained within special information repositories known as knowledge bases and are made publicly available either in the form of raw data files or via triple stores, which provide similar functionality to ordinary RDBMSs[4]. The underlying building block of a knowledge base is a subject-predicate-object triple, whereby subjects

[3] W3C. Resource Description Framework (RDF): `http://www.w3.org/RDF/`

[4] Wikipedia. Relational database management system (RDBMS): `http://en.wikipedia.org/wiki/Relational_database_management_system`

and objects are allowed to be interchanged. A subject is the identifier of a resource (an entity), a predicate the identifier of a relation, and the object is either the identifier of another resource or a concrete value, such as a String literal or some other primitive data value. A knowledge base is formally divided between a definition schema, comprising the terminological basis of the data, and the actual instance data providing an instantiation of the conceptual schema. One can conceptualise a knowledge base as a loosely coupled directed labeled graph (DLG), whereby subjects and objects are treated as nodes and predicates as labeled edges (relations) between them. As an example, Figure 2.1 shows an abstract RDF graph fragment (omitting namespaces and URIs) containing data about researchers, a project, and a sponsoring agency. DLGs are a common and generic model to describe possibly any type of semantic network or association graph. On the Semantic Web, relations are first-class URI resources and can be defined locally or reused from existing vocabularies.

2.2.2 Intrinsic technical problems

Dealing with searches over Semantic Web data raises many issues. The definition of a document in conventional IR is now projected onto entities that may be connected to a multitude, and possibly non-deterministic set, of object and datatype relations. Besides the semantic matching of keywords to ontology concepts and other RDF literals (a topic central to IR research in general, requiring disambiguation and expansion of polysemous words and phrases), a search process has to interpret and utilise the graph structure. Even for simple queries, a sparsely distributed network may require evidence to be traversed in the graph until an association with candidate resources can somehow be established. Query evidence may be connected to relevant objects, but not directly to the sought-for resources (for example, searching for “EnAKTing Researchers” or “EPSRC Researchers” in Figure 2.1). Depending on the indexing and search model utilised, this may be accomplished in various more or less successful ways. Additionally, the presence

of explicit semantics in the data uncovers functionality that can lead to more expressive query construction, essentially allowing queries of more complex graph patterns to match e.g. queries pertaining to multiple triple patterns with variable restrictions on types and attributes, such as "EnAKTing members who live in Massachusetts". Systems may opt to exploit this potential for hybrid or semi-structured query capabilities.

Key to success when dealing with ambiguous keyword queries remains the effectiveness of the ranking produced by a respective algorithm and its degree of portability, irrespective of any further processing incurred by a system. For example, an effective model restricted to a specific domain will face deficiencies when ported onto new datasets from the ever-increasing Web of data. In the same way, a very efficient and portable algorithm with severe deficiencies in its ranking cannot satisfy the expected utility of end users. In addition, a large-scale semantic search engine will have to cope with a very large and complex space of distributed knowledge bases on the Web, imposing hard scalability and performance restrictions. A good balance between effectiveness, efficiency and portability across domains is a necessary commitment for successful implementations.

2.3 Ranking Approaches

2.3.1 Query graph construction and exploration methods

Keyword query processing over graph-structured data has emerged as an important research topic in the wider field of database research. A considerable amount of research reported in the literature focuses on adapting keyword search to relational and XML databases, which can also be portrayed as graphs or trees. In this section, we look at various techniques that interpret keyword queries as substructures of a graph and apply various heuristics to estimate the relevance of each substructure. Our focus is on methods applied on Semantic Web data, although we start by looking at earlier works dealing with conventional databases. The two are in fact very similar and the former may appear

as extensions to earlier works.

Conceptually, databases can be regarded as graphs or trees, with nodes resembling tuples or XML elements and edges resembling foreign-key relations (w.r.t. relational databases) or element containments and IDREF/ID links (w.r.t. XML databases). Techniques that operate directly on XML data are very popular in the literature, although most depend on tree-structured data (Florescu et al., 2000; Guo et al., 2003; Cohen et al., 2003). In a typical scenario, an algorithm computes minimal cost connected trees as answers to a query. Techniques that focus on relational databases consider a graph orientation, and are thus more related to the Semantic Web, which is inherently graph-based.

2.3.1.1 Database techniques

There is a large body of work dealing with keyword searches inside databases. These are generally divided between schema agnostic techniques that operate directly on data and database extensions that require a database schema. Popular methods focus on finding a minimal subgraph/tree in the network that connects all the nodes matching the keyword elements. BANKS (Bhalotia et al., 2002), for instance, is a popular schema-agnostic architecture that employs a backward search algorithm starting from the nodes containing at least one query keyword and iteratively traverses incoming edges until a connecting answer root is reached. The answer to a query becomes a rooted directed Steiner tree (Dreyfus and Wagner, 1971) containing a directed path from the root to each keyword node. The model comprises a combination of relevance clues from nodes to edges, including heuristics to measure the prestige of nodes as a function of their in-degree and edge weights reflecting the strength of relationships (proximity) between tuples. Kacholia et al. (2005) propose an extension to BANKS considering bidirectional propagation factors e.g. methods to traverse the graph both backward from keyword nodes and forward from potential roots. This has the effect of finding more efficiently potential roots in the network - it was proven that fewer iterations were needed, hence the

model can deal with situations when query keywords match a very large number of nodes. As an extension to the above, the BLINKS framework (He et al., 2007) introduces a novel indexing scheme using block-based partitioning to improve the efficiency of bidirectional graph exploration. Similar popular approaches are presented as database extensions in DBXplorer (Sanjay et al., 2002) and DISCOVER (Hristidis and Papakonstantinou, 2002); these operate on the schema graph of databases, hence rely heavily on the database schema and the infrastructure of the underlying RDBMS.

2.3.1.2 Semantic Web techniques

Recent studies on the Semantic Web have been motivated by similar ideas. The general focus is on the computation of conjunctive queries from keywords using Semantic Web data. Zhou et al. (2007) explore a process for automatically translating keyword queries into formal logic queries via a prototype system known as SPARK. Given a keyword query, SPARK maps the keywords to various knowledge base constructs and outputs a ranked list of SPARQL equivalents, which the user can choose to execute. The process (illustrated in Figure 2.2) starts with keywords being enumerated into several combinations and mapped to resources in the knowledge base; a series of morphological and semantic processing steps (string comparisons and synonym expansion using the WordNet electronic lexicon) facilitate the mapping and assign a confidence value to each mapped keyword. The graph construction phase takes as input the mapped resources, splits them into different query sets via further enumeration, and applies a Minimum Spanning Tree algorithm to construct possible query graphs from each query set. The output query graphs are essentially a set of candidate SPARQL queries to be ranked before presented to the user.

Ranking in SPARK is driven by a combination of diagnostic probability estimates for each candidate formal query. Precisely, it is defined as

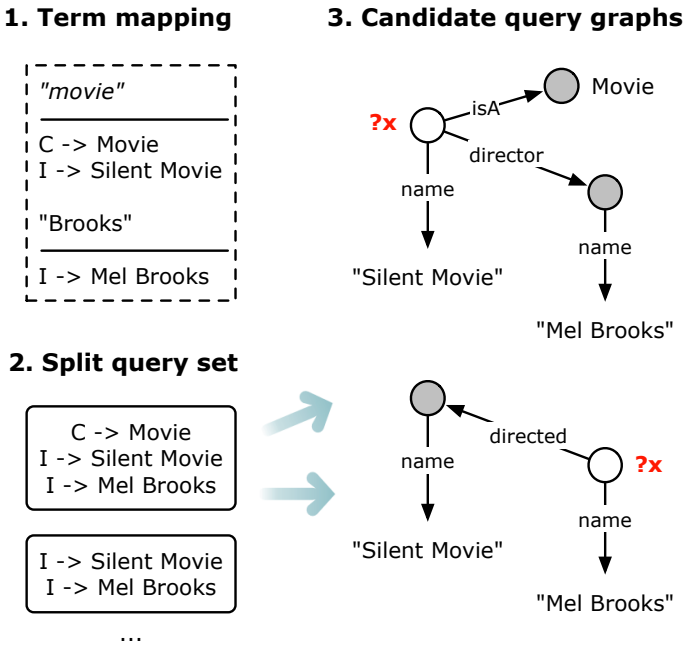


Figure 2.2: Query graph construction process: From simple keywords to a set of candidate logic queries. Simplified reproduction from (Zhou et al., 2007).

$$p(F|D,K) = p(F|K)p(F|D) \tag{2.1}$$

assuming independence between the relevance of formal query F to the knowledge base D and the keyword query K. The former, query diagnosis $p(F|K)$, incorporates the confidence values of each mapped keyword and the overlap of F with the original query. The likelihood $p(F|D)$ considers the information content of a formal query as a measure of the relative frequency of its relations as occurring in D. The model is flexible to parameterisation, offering users the option to adjust the ranking via a slider on a sigmoid function, such as to favour frequent vs infrequent relations. SPARK was evaluated on a set of manually constructed knowledge bases and translated queries from the Mooney Natural Language Learning Data. Results indicate that the model works best with medium or short sized queries (2-6 terms); more complex natural language

queries proved too ambiguous to understand and translate e.g. queries involving negation, superlative forms and other value constraints.

In a similar study, Tran et al. (2009, 2011) also extend the notion of query graph construction to answer sets that are not restricted to trees but that can be graphs in general. In this approach, keywords are interpreted as both vertices and edges to enable better reasoning with more complex queries e.g. "authors working at Stanford University that have won a Turing Award". The knowledge base is pre-indexed into an inverted index of keyword-element mappings and a *summary graph*, which captures relations between classes and instances into a graph index via type and subsumption information. The aim of a summary graph is to reduce the solution space to a more concise equivalent that can be used more efficiently for exploration. The rest of the process for top-k query computation is summarised in the following steps:

1. Mapping query keywords to elements of the data graph (literals associated with nodes and edges)
2. Exploring the data graph by traversing paths from the keywords to potential connecting elements
3. Merging paths that meet at connecting elements to construct a set of matching minimal subgraphs
4. Ranking matching subgraphs to produce a top-k query answer set

The computation process can result in multiple subgraphs corresponding to several possible interpretations of the keywords. Results from the process are effectively a set of matching structured queries, which the user can choose to execute and retrieve their answers individually. The relevance of computed queries, or subgraphs, is assessed via a combination of *cost functions*, defined as a monotonic aggregation of scores derived from the paths in a graph. More precisely, a cost function has the form

$$\sum_{p \in P} \sum_{n \in p} c(n) \tag{2.2}$$

where P is the set of paths in the answer graph and n is an element to be associated with a specific cost. The authors experiment with path lengths (favoring graphs with entities closer together), popularity scores (simple metrics to favor larger graphs), and keyword matching scores (incorporating both syntactic and semantic similarities using WordNet) to complete the functions for each graph. The precise implementation is based on the Threshold Algorithm, except lower bounds correspond to highest costs, and upper bounds to lowest costs. Experiments conducted over the DBLP[5] dataset concluded that keyword matching scores were the most prevalent factor with superior results in all cases. It remains unclear, however, whether combinations of cost functions were indeed assessed and what the best combination would be.

A closely related work, although still in its initial architectural stages, is presented by Parthasarathy et al. (2011). The authors also experiment with type and subsumption information, except this time exploited to traverse the data graph and construct an initial set of matching graphs after keywords are mapped to nodes and edges in the data graph. Henceforth, a set of pruning and hooking heuristics are introduced to merge subgraphs together. Pruning eliminates loosely hanging nodes, and everything that remains can potentially be mapped or merged across pairs of graphs. The outcome may be multiple answer graphs and ranking becomes essential to order the results. The authors consider heuristics to estimate the structural compactness of the elements in the output graphs, the textual relevance of keywords to the nodes mapped, and the relevance of nodes and edges. We refrain from further details since evaluation of the method has not yet been carried out.

2.3.2 Spreading Activation

Spreading Activation is a popular technique used traditionally in psychology to study human memory phenomena and operations, such as retention and recall of cognitive units

[5]DBLP Computer Science Bibliography: `http://www.informatik.uni-trier.de/~ley/db/`

of memory (Anderson, 1983). The framework has been widely adopted in other fields where semantic or associative networks are the primary form of knowledge representation. In Information Retrieval, Spreading Activation has been used for several applications (Crestani, 1997). The algorithm provides a basic inference solution to network data structures where concepts are treated as nodes and relationships as weighted or labelled arcs between them. The intuition is a fairly simple one: given an initial activation value for a set of nodes, Spreading Activation will traverse the network iteratively and spread the activation values to neighbouring nodes. There are possibly many different processing techniques, restrictions, and decay conditions that can be applied, but the general idea is that when propagation halts, each and every node in the network will be activated with a certain value.

In its basic form, we may define the input $I_j(t_{i+1})$ of node j at time t_{i+1} to be the sum of the outputs of the nodes that connect to it, weighted by the type of relation that holds between them:

$$I_j(t_{i+1}) = \sum_k O_k(t_i) \times w_{k,j} \times (1 - a), \quad a \in (0, 1) \tag{2.3}$$

where $O_k(t_i)$ is the output of node k and $w_{k,j}$ is the weight of the relation. It is quite common to associate a loss function a with the propagation, such that to give preference to shorter paths in the network. Spreading Activation conveys an attractive formalism for processing query evidence across Semantic Web networks. The following systems have both used Spreading Activation in a similar form to develop their inference processes. Different techniques are applied to associate weights with relations.

2.3.2.1 Examples of Spreading Activation use

OntoSearch (Jiang and Tan, 2006) is a unique prototype solution that combines ontology-based inference with classical keyword-based methods at query time for retrieval. Al-

though the method has been demonstrated on a collection of semantically enriched documents, the algorithm is conceptualised at the entity level, thus is hereby presented as an entity type of ranking model.

Resource URIs in OntoSearch correspond to instance entities and are treated as compound vectors of keywords and concepts. Keywords constitute the textual descriptions of resources (what would be equivalent to a label) and concepts assume taxonomical ontology classes (concepts related to resources via some type of instantiation edge). The method uses a TF-IDF (Baeza-Yates and Ribeiro-Neto, 1999) measure to assign weights to keywords and binary values to indicate a concept's association with the corresponding resource. Upon arrival of a query, the system uses the submitted query terms to retrieve an initial list of resources via a keyword-based search method. The concepts associated with the retrieved resources are then seeded into a Spreading Activation algorithm to infer more concepts that are semantically related to the original set. The outcome of the algorithm formulates a compound query vector with keywords and weighted concepts (concepts activated by spreading activation). OntoSearch utilises the *relative frequency* of properties to determine the weights used in Spreading Activation. Ranking is then facilitated by a straightforward dot product of the resource vectors (which remain intact) and the extended query. OntoSearch extends Spreading Activation with personalised views of a domain in the form of user ontologies encoding *relevance feedback* provided for past queries. These are factored into the concept weights, assuming a time decay factor based on the time interval between queries.

The OntoSearch method has been evaluated on a small collection of academic publications from the ACM Digital Library. The ACM Computing Classification System[6] terms were used to index the documents with taxonomical information, assuming the dataset's underlying ontology for the experiments. A user study was conducted to evaluate the system against a conventional keyword-based search engine (Lucene) and

[6]ACM Computing Classification System: `http://www.acm.org/class`

provide relevance assessments for the retrieved documents. Although no strong indication of the statistical significance of the tests is apparent, OntoSearch outperformed Lucene in terms of Average Precision on a set of 30 test queries. Performance appeared to be remarkably higher at low recall levels, while the two approaches naturally decreased and converged at high recall levels. Usage of a user ontology indicated improvements over the baseline method for $3/5$ users.

A very similar approach, combining Spreading Activation with traditional keyword processing, is presented by Rocha et al. (2004). One of the main ideas explored was how to extract information from the link structure of knowledge bases to associate weights with object relations. The authors combine two measures, namely *cluster* and *specificity*, and use a hybrid Spreading Activation technique that combines numerical weights with the labels of properties.

The *cluster* measure is treated as an asymmetric estimate and attempts to establish the degree of similarity between two related instances. The algorithm is a straight adaptation of the clustering function developed in (Chen and Ng, 1995; Chen and Lynch, 1992) for constructing association networks from term co-occurrence rates in documents. The measure interprets the similarity of two entities C_j and C_k as the ratio of their intersection with other entities in a knowledge base, relative to the event space of either of the two entities. Let N_{ij} denote the event that C_j is related to C_i, taking on values from the set $\{1, 0\}$ (indicating whether the event is $true$ or $false$), and N_{ijk} denote the event that C_j and C_k are both related to concept C_i, again taking on values from the set $\{1, 0\}$. Considering a knowledge base with n entities, the similarity of C_j and C_k relative to C_j is given as follows:

$$W(C_j, C_k) = \sum_{i=1}^{n} N_{ijk} \Big/ \sum_{i=1}^{n} N_{ij} \tag{2.4}$$

which is a probability estimate between 0 and 1. Note that the equation is asymmetric; switching the denominator to sum over N_{ik}, instead of N_{ij}, the intersection becomes

relative to C_k. This is the main characteristic of the *cluster* function and holds many implications on semantic or associative networks, where directed arcs establish connections between nodes.

The *specificity* measure is similar to the IDF convention and is used for discriminating against very common relations. The measure is inspired by the work of Stojanovic et al. (2003) on differentiating property instances based on their utility in knowledge bases. The specificity of a relation r between two instances C_j and C_k is given as:

$$W_r(C_j, C_k) = 1/\sqrt{n_k} \tag{2.5}$$

which is inversely proportional to the number of instances (n_k) that link to C_k via the given relation. The measure is asymmetric and interprets how specific the destination concept is. The result saturates over increasing values of n_k.

Combining weights with labelled arcs involves assigning additional manual weights to properties; hence Spreading Activation is extended in the aforementioned work with an extra weighted factor in each propagation. These can be relative weights used for fine-tuning propagation in a network e.g. zero-weighted properties can clamp a network and not allow propagation to flow through the edge, while higher weights can be associated with more important properties.

The ranking process is similar to OntoSearch. Results from an initial keyword-based search using Lucene are supplied to the Spreading Activation algorithm, and the initial ranking is used to define the activation values of nodes. The outcome may be a reordering or expansion of the initial results list or a new set of results altogether. There is no refactoring of query input after the Spreading Activation process halts (as done in OntoSearch). The proposed algorithm lacks empirical evaluation with a baseline method, but a qualitative analysis from domain experts indicated promising results on two separate implementations. It was observed that many relevant results would only have been possible through an otherwise complicated manual chaining of queries.

2.3.3 Classic probabilistic retrieval models

Probabilistic models in IR have been integral for reasoning with uncertainty in a wide range of tasks. Some of the earliest and pioneering techniques in the field were designed around models that base their core assumptions on rudimental probabilistic and Bayesian principles, such as the binary independence and language modelling approaches (Croft et al., 2009; Fuhr, 1992). Uncertainty is an intrinsic problem in IR. A major difference between IR systems and other information systems is the lack of query formulation that can represent uniquely an information need and a clear procedure to decide whether an object from a knowledge base is a correct answer. Probability theory has been the most well-studied paradigm for modelling solutions to IR, with the most successful frameworks serving as extensible solutions on which more complex models have evolved.

Any modern textbook on IR typically offers extensive coverage of probabilistic models, which can range from early principled approaches (dating from the early 1960's) to more abstract inference network models serving as generalisation frameworks. In this section, we present coverage of two retrieval models that have seen wide adoption in the literature and motivated recent experimental developments in Semantic Web search.

2.3.3.1 Language Model

Language Models are a general formal approach to IR, with many variant realisations (Zhai, 2008; Croft et al., 2009). In their most common use they are known as query likelihood models, where the definition stems from the use of probabilistic reasoning to measure the likelihood that a query can materialise given a document specification. Effectively, the method associates a probability distribution over the occurrence of words in the index vocabulary of a collection. A document specification becomes a sampling of words from the distribution and the goal is to measure how likely it is that a document is about the same topic as the query. Language Models provide a generic Bayesian interpretation to the relevance of queries and documents, which has the general form:

$$p(D|Q) \propto p(Q|D)\, p(D) \tag{2.6}$$

for a query Q and a document D. The likelihood that a query is relevant to a document is usually treated with naive term independence assumptions, as in:

$$p(Q|D) = \prod_{w \in Q} p(w|D) \tag{2.7}$$

while the document prior is seen as a useful parameter for introducing additional criteria to favour documents with special features. The diagnostic support accorded to a document by a single query term $p(w_i|D)$ is commonly associated with a Dirichlet smoothing estimation, as in:

$$p(w_i|D) = (1 - \lambda_i)\, p(w_i|C) + \lambda_i\, p(w_i|D) \tag{2.8}$$

where $p(w_i|D)$ and $p(w_i|C)$ usually translate to the relative frequency of term w_i in document D and across the entire collection C. The smoothing parameters $\lambda \in (0, 1)$ are usually constant to the current document. Language Models typically associate separate probability distributions to queries and documents and the Kullback-Leibler (KL) divergence is used to compare the two models in terms of how close they are to each other i.e. the relative entropy or information gain from one to the other. Documents (or result graphs) can then be ranked in increasing order of the KL divergence. Assuming P_Q and P_D to be the probability (likelihood) distributions associated with a query and a document respectively, the KL divergence is given as:

$$KL(Q||D) = \sum_i P_Q(T_i)\, log \frac{P_Q(T_i)}{P_D(T_i)} \tag{2.9}$$

where the probability distributions are over the set $T_i = \{t_1, ..., t_n\}$ of n terms in the corpus.

Language Models are in general based on a very intuitive and extensible framework. The specification of a document is masked by a simple aggregation of individual scores (equation 2.7 - avoiding complex term co-dependencies), which are adjusted on a finer scale by weighting different sources of information (equation 2.8 - collection-wide and document-centric information). This serves as an interesting formalism for wider adoption of the model.

Elbassuoni et al. (2009) investigate the use of a Language Model to rank results to triple-based query patterns, whereby queries are treated as either purely structured or keyword-augmented triple patterns. The method fundamentally extends the notion of documents in traditional IR to a large all-encompassing graph of triples. A query Q is treated as an n-triple pattern (or relaxed pattern with variable predicate matches) and any subgraph of n triples from the knowledge base is considered a potential result-graph to the query (essentially assuming the role of a document in traditional IR). The method uses the relative frequency of individual triples (as opposed to terms) to approximate their marginal likelihood contribution in both exact triple matches and keyword-augmented queries. The authors refer to this as the relative *witness count* of triples. They do not appear to account for the within-triple frequency of terms, which in turn is surprising, given that frequency values are accounted for in the outlined keyword indexes. Realistically, datatype relations can be associated with more verbose literals, such as the case of labels and descriptions. Term frequency (e.g. in terms of TF-IDF) is important to differentiate the eliteness of resources to those terms.

The proposed method was suitably evaluated on two datasets and benchmarked against three other approaches. The experiments were based on a subset of IMDb[7] and LibraryThing[8] (a catalogue and forum of books), and the competitors included the Web Object Retrieval (WOR) method (Nie et al., 2007), BANKS (Kacholia et al., 2005), and NAGA (Kasneci et al., 2008). These are similar methods that operate on structured data

[7] Internet Movie Database: `http://imdb.org`
[8] LibraryThing: `http://www.librarything.com`

at the entity level and use different types of graph analytics to rank results. Evaluations were based on a user study to estimate the relevance of the results produced by each of the contestants. The outlined methodology outperformed the other methods on both datasets in terms of NDCG (Croft et al., 2009). It remains unclear, however, whether the proposed strategy can operate effectively over crisp RDF-centric knowledge bases. The authors used a customary search engine to approximate the *witness counts* of triples, as required by their model. Consequently, a ranking tied to an external search engine may not be durable to self-contained RDF knowledge bases where triples are expected to be distinct.

Balog et al. (2011) have used the Language Model as part of their competing system at the Semantic Search Challenge in 2011. Their main experiments involved an extension to the model to contribute field-level scores to the representation of the entities being evaluated. This was fairly straightforward to achieve, given the vague specification of term probabilities in the model. The individual scores of terms were projected onto field-specific dependencies adjusted by a prior score reflecting the importance of each field considered (f):

$$p(w|D) = \sum_{f \in F} p(w|D_f)p(f) \tag{2.10}$$

The individual term probabilities were then smoothed by Dirichlet priors as normal, except using field-specific and entity-level information; more specifically, functions to incorporate the length of each field being considered and field/entity-specific background models. The authors further explored propagation heuristics to communicate the individual scores of entities to connected entities via *sameAs* relations extracted from DBPedia[9].

Similarly, the authors of WOR (Nie et al., 2007) applied the Language Model at the level of Web objects, whereby an object was defined as a collection of database records of

[9]DBPedia: `http://dbpedia.org`

multiple attributes/fields aggregated from multiple Web sources. The authors experiment with variations of the model based on different levels of granularity of objects. In their best approach, individual term probabilities were extended by an additional dimension, incorporating the various possible object representations from multiple sources/records:

$$p(w|D) = \sum_{r} \sum_{f \in r} p(w|D_{r,f}) p(f)\ p(r) \tag{2.11}$$

where the prior of individual fields $p(f)$ was treated as a smoothing function incorporating the importance of the field and the accuracy of the field extraction phase. Similarly, the prior of a record representation $p(r)$ was used to incorporate the accuracy of record detection.

2.3.3.2 BM25F

BM25F (Robertson and Zaragoza, 2009; Robertson et al., 2004; Jones et al., 2000) is a state-of-the-art technique for structured document retrieval. The method was originally conceived in 1976 as a simple probabilistic model, known as the Binary Independence Retrieval (BIR) model, and was chiefly designed to integrate user feedback information into a ranking formalism. The original assumption was that documents could be classified between relevant and non-relevant sets and that terms are distributed differently within the two sets. In the absence of relevance information, the model encloses a ranking function that works similarly to a TF-IDF hybrid, in the sense of adopting collection-wide and document-centric term occurrence statistics. The BIR model, also known as "Okapi BM25", was later extended to manage structured document retrieval (in particular, Robertson et al. (2004) formalised the method in 2004), by extending its ranking functions to multiple weighted fields as opposed to flat documents e.g. by weighing occurrences of terms in the *title*, *body*, or *anchor text* of Web pages. In general, the newest version of the model, BM25F, is known to improve retrieval effectiveness by

use of non-linear frequency saturation functions, document and field length normalisation, and field weights for structured IR. This entails a rather lengthy list of tuning parameters. In particular, $2K + 1$ parameters for K fields need to be estimated per collection for the model to reach its optimum potential. Parameter optimisation in BM25F is a heavy experimental process, requiring training datasets with possibly large volumes of queries and assessments.

BM25F is mostly known in the literature as a precise ranking function and not an extensible framework as the case would more naturally be for the Language Model. A few recent studies on Semantic Search have used BM25F for entity-oriented search, whereby Semantic Web resources have been explored primarily at the level of datatype information.

Pérez-Agüera et al. (2010) designed an experiment whereby entity resources are generalised as structured documents consisted of five fields: all text from property values, words from the URI of the entity, words from the URIs of objects (associated entities), words from predicates used to link to the entity, and words from the URI of associated classes via *rdf:type* relations. The categories were weighted with individual field boost factors while the remaining parameters were assigned values guided by the the authors' judgement. The experiments primarily aimed to recap shortcomings of techniques that fail to implement correctly saturation effects and field-weighting, therefore demonstrate how BM25F can address the context correctly. The authors used the 2009 INEX Wikipedia collection for evaluation, in turn transposed to RDF by mapping to equivalent DBpedia entries. A series of Precision metrics were employed to compare BM25 and BM25F with corresponding variants of the Lucene engine. The results exhibited quality improvements over all the test beds using the BM25 variants. Lucene appeared to perform significantly worse when structure was taken into account. This is certainly indicative of the method's shortcomings in dealing with document structure.

In a similar study, Blanco et al. (2011) designed an experiment with BM25F over the

Billion Triples Challenge 2009 dataset, the dataset used as part of the Semantic Search Challenge in 2010/11. In this experiment, the authors capture datatype information from the top-300 datatype properties in the collection and assign different weights to different categories of predicates. Properties were manually classified into three classes (important, unimportant, neutral) and weights were assigned for each class. Domain names were also classified between important and unimportant, with `dbpedia.org` and `netflix.com` constituting the important category. A simplified version of BM25F was used where individual field lengths were projected onto a higher dimension as the size of the enclosing entity, effectively reducing the index space required to store individual field lengths for a potentially very large set of entities and property values. The method appears to be a revised version of the winning team's submission at the 2010 Semantic Search Challenge (Blanco et al., 2010). Results from the experiments indicated 42% improvement in average precision over the best run at the 2010 competition.

2.3.4 Link-analysis inspired methods

The hypertextual structure of the Web has been one of the richest sources of information for developing reliable ranking heuristics. There are conceivably many applications that can benefit from analysis of hypertext links, including document classification and clustering, deciding what pages to crawl, prioritising documents in vast posting lists and composite scoring of web pages on any given query. The two most popular contributions in this area with important implication on Web search have been the HITS algorithm by Kleinberg (1999) and the PageRank algorithm by Brin and Page (1998) (see also Page et al., 1999). The former is typically treated as a query-dependent algorithm, useful for such cases as finding communities of practise on a given topic or post-query processing and sorting of documents. PageRank is most commonly known for query-independent or prior scoring of documents, providing a static score element for Web pages on which to base a notion of importance or popularity. Both algorithms are iterative algorithms

whose values are expected to converge after a certain number of iterations.

Semantic Web data is in many ways similar to the hypertext Web, in that links constitute a fundamental notion of relevance. However, resources on the Semantic Web can be related via a multitude of heterogeneous links, each indicating a different type of association. For this, static scoring via conventional link analysis to derive scores of popularity or importance demands deeper elucidation of what is actually being conferred across Web resources. The PageRank algorithm, primarily due to its popular pose in the literature and as part of the Google search engine, has served as a common baseline for link analysis on Semantic Web graphs.

2.3.4.1 PageRank

PageRank assumes a homogeneous structure of the Web, whereby links are assumed to carry a uniform endorsement to the analysis of pages. PageRank has a simple intuitive probabilistic interpretation that tries to emulate the likelihood of a person randomly surfing the Web to arrive at a particular page. The PageRank of a page is derived from its backlinks and is proportional to the sum of the ranks of all the pages that link to it. If we assume x to be a page on the Web, B_x to be the set of all pages that link to x, and N_x the total outgoing links of x, PageRank is computed as follows:

$$R(x) = c \sum_{y \in B_x} \frac{R(y)}{N_y} + E(x) \tag{2.12}$$

where c and $E(x)$ are treated as normalising constants ranging between 0 and 1 and are used to balance the equation. c indicates the maximum rank contribution of the set of pages B_x and $E(x)$ adjusts the score to an upper limit of 1, while setting a uniform initial value across all pages. Given the algebraic relation of the two parameters, they are often expressed as d and $(1 - d)$ resp. Given the above formulation, the importance that a page confers to x is determined by the importance of the page itself and is inversely

proportional to the number of pages that it links to.

Extensions to PageRank for weighted link analysis are a common scenario in the IR and database literature. A reflective example is Microsoft's PopRank model (Nie et al., 2005), which adopts the algorithm to "popularity propagation factors" learned from partial ranking lists via a machine learning approach. The method emulates PageRank's "random surfer" model to a "random object finder" and has been applied successfully on large document collections. ObjectRank (Balmin et al., 2004), another example, applies PageRank in a query-dependent fashion to satisfy keyword searches in databases. The technique assumes a weighted schema graph with links assigned different authority transfer rates. XRank (Guo et al., 2003) is a similar approach for XML classification. The following examples constitute reflective uses of PageRank for ranking Semantic Web data for search.

2.3.4.2 Uses of PageRank for Semantic Web search

Some of the earliest retrieval techniques applied on the Semantic Web focused on finding relevant ontologies, or Semantic Web Documents (SWD), as potential matches to a customary set of keywords. Effectively, a general methodology for ranking SWDs can work for ranking RDF instances or entities, but usually invested approaches are not always that generic. **Swoogle**[10] (Ding et al., 2005, 2004b) dominated this area of development, maintaining a robust index to ontologies across a wide range of domains. The main construct of Swoogle's ranking is based on a modular weighted PageRank (OntoRank) that aims to assess the popularity of documents by exploring different inter-document relations. These take the form of axiomatic referral links, such as when a SWD uses or extends vocabulary terms defined in another (for example, via *rdfs:subClassOf* or *rdf:type* relations). The main extension to the original algorithm involves the inclusion of manually-specified navigation preferences, which take the form of weights assigned

[10]Swoogle: `http://swoogle.umbc.edu/`

to the semantic links between documents. Considering $link(y, l, x)$ to denote a relation l between x and y and $weight(l)$ to be the user specified weight for the given relation, then PageRank is adjusted as follows:

$$R'(x) = d \sum_{y \in B_x} \frac{R'(y) \times f(y,x)}{\sum_{link(y,\_,n)} f(y,n)} + (1-d) \tag{2.13}$$

$$\text{where} \quad f(y,x) = \sum_{link(y,l,x)} weight(l) \tag{2.14}$$

$f(y,x)$ is the aggregated weight over all the relations from y to x. OntoRank further accumulates a document's final score with the ranks of all the documents that import the respective ontology via *owl:imports*. Swoogle's ranking is inclusive and OntoRank is also used to provide ranking for ontology terms in a knowledge base e.g. facilitate retrieval of properties and classes based on how often they are used and the popularity of the documents that use them. The main pivot of the approach is whether the underlying documents are well connected or cross-referenced, which may not necessarily be the case. Evidently, autonomous documents may end up receiving poor OntoRank scores, a case that has been addressed more precisely in (Alani et al., 2006).

SWRank (Wu and Li, 2007) is a prototype entity-rank method that, like Swoogle's OntoRank, explores the use of multiple relations between resources to implement PageRank-like analysis. SWRank considers overall *hub* score to be the popularity of an entity, which is reverse to conventional PageRank. The approach works by reversing the direction of all the edges in a RDF graph and applying weighted PageRank (as with Equation 2.13) on the reversed graph. The outcome is a shift of orientation but yet with relative consistency to the original algorithm. *Reverse PageRank* is a speculative technique for hypertext browsing and has been investigated previously by Fogaras (2003). SWRank works consistently across the schema and data levels of a knowledge base, hence involves no pragmatic differentiation between schema and assertional semantics.

The outlined system in (Wu and Li, 2007) combines SWRank with classic vector-based ranking for overall retrieval of entities. The vector-based scheme emulates traditional TF-IDF on all the literal values associated with resources. A resource is effectively treated as a bag-of-words, without further processing of datatype relations. Experiments on datasets generated from SourceForge[11] and SchemaWeb[12] revealed comparable convergence speeds between SWRank and plain PageRank. SWRank is also indicated to coincide more with the "Project Web Hits" statistics from SourceForge, a rather promising outcome. The main caveat we observe with reversing the algorithm is that the orientation is shifted from distilling authorities to focusing on hubs in the network. Traditional PageRank would classify a resource as popular if many other resources link to it and not many others, and many resources link to them and not many others, which is a reasonable assumption. With SWRank, it appears that resources are classified as popular if they link to very few resources that link to very few others; such implies a "close" community finder rather than a popularity estimate. The motivation of using Reverse PageRank needs deeper justification, especially when employed as a general algorithm for enhancing the ranks of resources.

Sindice[13] (Oren et al., 2008; Tummarello et al., 2007; Delbru et al., 2010a) is an end-to-end search engine for Linked Data on the Web, offering a suite of API tools for querying the indexed sources (at the time limited to keyword, URI, and Inverse Functional Property lookups). The engine underlying the keyword lookup processor (SIREn[14]) extends on the Apache Lucene project and supports full-text and semi-structured queries. Sindice employs a two-layer hierarchical link analysis model to rank resources, known as DING (Dataset rankING) (Delbru et al., 2010b), that distinguishes between entity and dataset information. This is illustrated in Figure 2.3. Links are aggregated from the entire graph and weighted as bundles of links and linksets via a linear TF-IDF inspired

[11] SourceForge: `http://sourceforge.net`
[12] SchemaWeb: `http://schemaweb.info`
[13] Sindice: `http://sindice.org`
[14] Semantic Information Retrieval Engine (SIREn): `http://siren.sindice.com/`

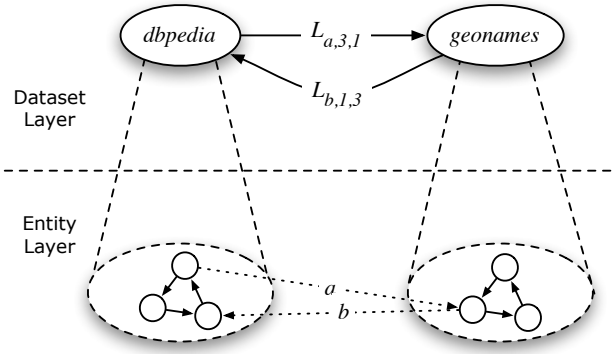


Figure 2.3: Sindice's two-layer model. Dataset Layer made up of inter-dataset link sets, and Entity Layer made up of inter and intra dataset links. Simplified reproduction from (Delbru et al., 2010b).

unsupervised method. The weighting scheme assigns a higher degree of importance to links with a high frequency in individual datasets and lower frequency across the entire dataset collection. The DING algorithm is an extension to PageRank (works exactly like a weighted PageRank where applicable) and defuses the weights into dataset and entity ranks by traversing the weighted graphs. The aim is to estimate the importance of datasets across the entire collection and that of entities on a per-dataset level. The final score is a linear combination of the two weights after normalising the ranks by the size of the datasets.

Sindice employs a variety of interesting methods to rank resources, but very little evaluation exists to demonstrate the quality of the approach, especially at different granularity levels of the algorithm. Experiments were conducted to evaluate individual parts of DING against a baseline method (operating on the full data graph). These revealed close correlation between the different methods, demonstrating that a global entity rank can possibly be interpolated via less expensive local computations. User studies were also conducted to assess the performance of the ranking on different datasets, using a similar methodology. Yet again, close correlation was found between the different components of DING and the baseline method.

SWSE[15] (Hogan et al., 2011; Harth et al., 2007) is another prototype data aggregation project that indexes Semantic Web data for searching. SWSE crawls and bundles RDF data with non-RDF sources (HTML documents, RSS feeds), and arranges the content into canonical bundles after analysing *owl:sameAs* and Inverse Functional Property relations. TBox reasoning is also adopted to infer new statements about the data. Ranking in SWSE is based chiefly around the notion of a Naming Authority (Harth et al., 2009), which aims to distinguish and establish a connection between an entity identifier (URI) and the source with the authority to assign the identifier, also referred to as Pay Level Domain e.g. `example.com` for `foo.example.com`. In the case of HTTP 303 redirections (a common scenario explored in publication of Linked Data) the Naming Authority is extracted from the redirected URI. Having constructed a Naming Authority graph, PageRank is used to derive scores for each source-level identifier. Property and *rdf:type* object-position URIs are assumed to be potent for over inflating the ranks and are not used in the derivation of the graphs. The rank of individual entity identifiers subsumes the ranks of the sources in which the identifier occurs. The intuition is that the more highly ranked the source mentioning a URI, the higher the rank of the term should be. SWSE combines the PageRank scores of URIs with simple TF-IDF query-dependent scores for overall ranking. There is no evidence of datatype property demarcation, although some indication is given that labels (literals linked to *rdfs:label*) are preferred over other primitive data values.

Evaluation of the Naming Authority strategy mainly focused on evaluating different variants of the algorithm (differing according to the level of the naming authority) and contrasting the results with a baseline method. This is similar to the Sindice experiments. The baseline method included a naive version of PageRank operating directly on entities by not taking sources into account. Experiments were conducted on several datasets, including a stripped version of the 2008 Billion Triples Challenge dataset. Quality

[15] Semantic Web Search Engine (SWSE): `http://swse.org/`

evaluation was driven by a user study, where 15 participants were asked to rate results from different top-10 ranked lists. The proposed algorithm exhibited improvements over the baseline method. Performance evaluations indicated similar runtime properties to the baseline method.

2.4 Conclusions and Future Research Directions

In this chapter, we have presented various techniques from the literature on semantic search involving ranked keyword search over graph-based Semantic Web data. We have presented important concepts and common techniques in some detail, which should appeal to readers interested in a deeper perspective over the various methods and systems implemented. As natural, this comes at the expense of a more complete survey over work in this area. In the following, we revisit highlights from the survey and outline key areas that future research may be directed. In the understanding that the material presented are but a small subset of a much broader theme, many of the longitudinal challenges in IR are effectively carried over. Directions for future work are deliberated on aspects that are mostly reflective to the course of the survey, whilst topics are by no means exhaustive.

2.4.1 Unifying ranking models

Ranking is among the most functional issues in search engines. A number of approaches have been presented in this chapter, but none of them stands out as the definitive solution. The evaluation processes outlined lack a common evaluation benchmark and procedure to be able to compare these models over the same dataset and queries. It is therefore difficult, if at all possible, to emphasise and contrast the features of the various techniques, while avoiding speculation. Furthermore, the different orientations of the techniques presented aim to address different aspects of a retrieval process. For example, link analysis techniques, as in the popularity-based measures presented, give us insight

on the linkage and density of the graphs surrounding entities. Propagation and graph exploration techniques are useful for distributing query evidence to the graphs. At the same time, probabilistic models, as in BM25F, have proven very successful in modelling and reasoning with different frames of contextual content in knowledge bases.

Given the diverse orientation of the various techniques, the question remains whether an overall view can be synthesised from such and other vertical approaches. Research on combining multiple models of relevance, therefore, seems highly relevant. Frameworks that can blend together query-independent with query-dependent techniques to prioritise query evidence across clusters of high proximity nodes or describe both probabilistic and logical processes (e.g. restrictions on types and predicates) to enable more complex constraint queries, may be a way forward. For example, methods that engage in query graph construction and exploration can potentially be combined with query-independent link analysis to equate the centrality of the graphs in the measurements.

Graphical models (Bayesian and Neural Networks) have been utilised successfully in IR as generalisation frameworks for combining distinct sources of evidence (relevance feedback, link analysis, and structure) to support the rank of documents (Croft et al., 2009; Baeza-Yates and Ribeiro-Neto, 1999). Pioneering techniques remained among the most competitive approaches throughout many years of IR research. These methodologies can provide insight on the development of clean formalisms for exposing both the structure and connectivity of resources to the statistical reasoning required to resolve ambiguous keyword queries over semantic data graphs. The idea of projecting data graphs to a *generative* probabilistic model (Bishop et al., 2006) to enable complex computations via graph manipulations is precisely what we propose in this thesis. A Bayesian Network approach to entity search over semantic web graphs can provide useful features for utilising the available semantics to answer various forms of user queries, while exposing a clearly-defined sample space to extend the ranking process with additional forms of reasoning e.g. link analysis of entities and prioritisation/demarcation of relations via

their usage and popularity in the knowledge base. In the following chapter, we introduce Bayesian Networks and their relevance in IR and proceed with the specification of the proposed model for entity-oriented search.

A number of systems outlined in the survey also make effective use of precise axiomatic relations to enhance the solution space during or prior to query processing. For example, Tran et al. (2009) explore type and subsumption information to develop summary graphs for more efficient graph exploration, and Hogan et al. (2011) explore OWL semantics to expand the solution space with additional explicit semantics prior to query processing. Similarly, Balog et al. (2011) explore *owl:sameAs* relations as a means to propagate query evidence across data instances. These are interesting operations that make effective use of some of the unique characteristics of Semantic Web data. Class and identity correspondences are among the most common forms of mappings on the Web of Data. We expect that the exploitation of these and other emerging common constructs to remain key in demonstrating how consensus and improved ranking can be achieved across heterogeneous data.

2.4.2 Indexing schemes

Retrieval efficiency is a major consideration when thinking about functional models to be used across a wide range of data collections. Although search engines generally do not have the costs associated with relational and RDF databases, there are significant obstacles in terms of fast response, since query terms may appear in a very large number of documents/entities that are associated with many other terms. The efficiency of a ranking model is largely dependent on the choice of an appropriate scheme to store and retrieve the necessary information. In conventional IR, inverted indexes (Zobel and Moffat, 2006) have been the most common structure explored and implemented across a number of standard search engine libraries. In an inverted index, vocabulary terms are typically the basic index unit stored in a dictionary (either a hashmap or tree) with

pointers to associated lists of document identifiers and relevant frequency information. Inverted indexes are a flexible data structure and extensions have been common in the literature, such as parametric zone indexes for models that distinguish between various parts of a document or positional indexes for models that prioritise phrases in text.

With respect to the semantic search methods reviewed in this chapter, there are a number of considerations that make the process rather unique from more conventional retrieval practises. First is the ability to retrieve resources based on words that appear in the values of properties (as in models that group property values into different weighted zones or expect restrictions on the names of properties) and the need to reason with object-level semantics for graph exploration and propagation techniques. Inverted indexes remain a natural course for modelling these types of associations. Most systems use or extend a standard search engine library (e.g. Lucene, MG4J, Lemur) to associate keyword-level indexes with entities, although, beyond a few exceptions, not much information is given on the precise implementation details.

Blanco et al. (2011) explore MG4J's positional indexes to expand terms with field information corresponding to the top-300 datatype properties from the Billion Triples Challenge dataset. The authors focus on an efficient implementation by exploring an additional index to group properties into three broader weighted classes, effectively leaving them with only three fields to parameterise each individual term. Tran et al. (2009) focus on an expressive keyword index for graph exploration. The authors use inverted indexes to associate terms to lists of connected nodes via specific predicates/edges and to the labels of edges and classes. To reduce time and space complexity, object-level semantics are captured between classes of entities as a summary graph index, hence instance-level relations are aggregated at a higher dimension. In a similar context, the authors of BLINKS (He et al., 2007) present a block-based partitioning scheme that divides the graph into several subgraphs and captures keyword-node, node-keyword and block-level proximity information into a set of inverted indexes for use in bidirectional

graph exploration. Both of the aforementioned experiments were elaborated on single datasets (largest ranging at approximately 26M triples) and illustrated affordable use for practical implementations.

Indexing schemes to support models over large and possibly multi-dataset environments will remain a key factor in future implementations. Whether standard libraries can be developed or extended to provide basic means for graph partitioning, propagation, and various levels of parametric indexes is a highly desirable prospect, since results can be enjoyed by the wider community without needing to re-invent aspects proven to work. Associated costs to index maintenance, combination of models (e.g. BM25F with graph exploration) and support for extended queries are interesting areas to explore as well.

2.4.3 Tasks, datasets, and evaluation

For many years, research in IR has been driven by careful and thorough evaluation of the quality of proposed innovations. Conference series such as TREC and INEX contributed to a community consensus on a portfolio of principled evaluation measures for assessing the performance of search algorithms. Methodical evaluation is key to making progress in the field. It is also essential to understanding if a search engine is being used effectively and if it provides the functionality it was conceived for.

Starting an evaluation campaign for semantic search is, however, far from trivial. The community will need to agree on a precise perimeter of queries to assess and a set of datasets that are mostly reflective to the context of search. The Semantic Web community has recently organised an "ad-hoc object retrieval" task (Halpin et al., 2010; Pound et al., 2010), which is a step in the right direction, providing a general reference RDF collection for entity-oriented searches. The collection focuses on Web queries (simple keyword queries) and a general sizeable corpus representative of real-world data crawled from multiple sources on the Web. An issue that may require further consideration is the precedence of selected domains in the collection (data and assessments), with

dbpedia.org taking more than 50% of the distribution. Some of the systems competing at the Semantic Search Challenge appear to have exploited the distribution for a better chance of winning (Halpin et al., 2010).

Results from the two consecutive runs of the Semantic Search Challenge are a good point of reference for comparison against a baseline of methods over the billion triples dataset. Arguably, some of the competing systems suffered from a rather conservative perspective, but a few systems (of which some reviewed in this survey) are interesting assimilations of popular techniques. An interesting next frontier may be the proliferation of different tasks to direct focus on specific application needs and enduring trends. For example, a task focusing on semantic-oriented queries (e.g. queries involving variable matching and restrictions on attributes) as opposed to plain keywords, or a task focusing on statistical and geographical data as found in the abundance of government-released datasets. Platforms that demonstrate good performance across a variety of domains will without doubt be key indicators to successful implementations, but a more gradual evolution from micro experiments to macro settings may be a more appropriate path. The community can then look forward to unifying the most competent solutions, those most appropriate to deal with the unique characteristics of each task.

2.4.4 Integration with user interfaces

In this review, we have chosen to focus on a single mode of user interaction and presented in detail several forms of algorithmic approaches for distilling information from knowledge bases to satisfy user queries. From a broader perspective, however, semantic search is very commonly viewed as an iterative and exploratory process in which the user can actively engage with the system via various forms of interaction (Uren et al., 2007; Hildebrand et al., 2007). The idea is to help the user explore the domain, find out what is there and construct complex queries from possibly several atomic or incremental operations. An interesting direction for future research is how to manage the integration

of end-user support utilities, such as multi-facet views, class menus and visualisation graphs, auto-complete functionality, and pre/post-query disambiguation components with ranking heuristics to accomplish more comprehensive and multimodal design models. For example, general frameworks that can blend together a set of best practises to support hybrid or semi-structured query generation, pre and post-query disambiguation, profiling of users and possibly retainment of context across sessions. Research into cognitive aspects is important in this context, such as how much interaction a user is willing to bear to improve her search results. Development of mature, off-the-shelf components that can be adapted readily atop of existing knowledge base stores or search engine libraries is certainly an attractive prospect.

Chapter 3

A Bayesian Network Model for Entity-Oriented Search

3.1 Introduction

In Chapters 1 & 2, we reviewed a number of developments in the field of semantic search, with emphasis on ranking models used to support search over graph-based Semantic Web knowledge bases. Some of the strategies outlined attempt to increase the effectiveness of retrieval by dedicating separate methods for the evaluation of textual information and analysis of the link structure in knowledge bases. Other methods exploit the graph structure as a means of propagation or graph exploration after instantiating a set of nodes from evidence in the query. Other methods adapt conventional techniques for structured-document retrieval and treat property values as separate frames of knowledge suitable for reasoning with probabilistic techniques (BM25F and the Language Model).

The various works presented encompass promising outcomes, but to our impression very few of them make a decent attempt to develop new and formal models to cope with the semantic structure of the data. Most of the approaches (with perhaps the exception of graph exploration techniques) are habitually based on conventional techniques in IR, which signal a rather conservative scientific development. Conventional IR techniques have been developed with a simple language model in mind (a mostly unstructured

collection of terms associated directly to documents). Semantic data, as discussed in Chapter 1 (Section 1.3.1), can entail a more complex model, one that does not identify clearly the textual representation of entities. Relevant information (terms) needed to resolve textual queries can be dispersed across the literal values associated with different entities, which can in turn be associated with the sought-for resources directly or via other intermediate entities. A formal model for entity search would make explicit this form of interconnection of entities without requiring pre-processing or pre-aggregation of values to suit the inner-workings of a more conventional approach e.g. aggregating all the textual information potentially related to a resource and then treating the resource as a customary document. Nonetheless, the range of techniques presented illustrate that many possibilities are possible, especially since we are dealing with a new form of data that has only recently started to be incorporated in the major research venues in IR and elsewhere.

In this chapter, we present the ground architectural components of a new retrieval model for entity search. The model attempts to expose new means of reasoning[1] with the link structure of Semantic Web data and offer better possibilities for end-users. The orientation of the model is on fully or semi-automatic query processing (involving free-form and semi-structured queries), inline with the general scope of the mainstream research on semantic search. The model integrates, into a single framework, link and content-based information available in knowledge bases, in a way that its semantics are clearly distinguished and differentiated in the retrieval process. We achieve this by developing a generative Bayesian Network (BN) model that is capable to express the explicit semantics associated with semantic resources and expose them to statistical scrutiny and inference procedures. The model is flexible and generic enough to be adapted to any type of URI, or otherwise, entity resource. There does exist an upper-

[1] As mentioned previously, "reasoning" here has a more literal meaning than its figurative sense in Semantic Web research, as in "DL Reasoning". We will occasionally use the term in this chapter to refer to probabilistic and statistical reasoning, such as Bayesian inference.

bound on the level of reasoning that we expect the model to achieve, but the model is flexible enough to accommodate extensions and deliberate reasoning with a wide range of assertional semantics.

The model is motivated from the Bayesian Network approach in IR and, as customary with similar approaches in IR, tries to generalise into a single computational framework the necessary constructs to reason with several sources of available knowledge. The model differs from similar deployments of BNs in IR in that it aims to represent, and make explicit in the inference process, the presence of multiple relations that potentially link semantic resources together or with primitive data values, as it is customary with SW data. This leads to a number of possibilities for exploiting semantics to enable new means of reasoning. Part of our goal in designing this model has been to enable reasoning with more complex or expressive information needs, with semantics specified explicitly by users or incorporated via more implicit bindings.

3.1.1 Key features of the model

This is a good time to highlight the distinctive characteristics of the model, before we delve further into its precise specification. To summarise, the model adopts three distinctive attributes in its ranking procedure:

1. Relevance/quality propagation
2. Unsupervised link weighting: object property demarcation
3. Expressive query modelling: mixing facts (implicit or explicit) with text

The three attributes, primarily 1 and 2, are necessary constructs to reason with data that pertains to directed labelled graphs, since query evidence can be associated with any node on the graph not necessarily candidate for retrieval (we have offered a motivating example of such a scenario in the previous chapter). The latter (attribute 3) will be accomplished via a process of external parameterisation, which works by embodying (either explicitly or implicitly) weighted logical conditions into the core propagation

processes encompassing the network model. This is the key feature of the model we are about to present: its ability to exploit the semantics of data in order to reason with queries of relative complexity, from simple keyword queries, such as names of people, events and organisations, to more expressive queries involving relations e.g. "colleagues of Jim Smith who live in California". The latter differentiates the model from the encompassing literature, which has largely focused on simple free-form queries, as seen in the various evaluation and competition venues employed so far. The model employs a variety of techniques to leverage the available semantics in the data (mostly focusing on interrelations between data items) to bring together a unified ranking procedure with a sound mathematical foundation and potential for further extensions and modifications.

The model is not necessarily restricted to SW data and may be applicable to any form of data that pertains to the triple-based representation of knowledge bases. The ground foundations of the model offer a rich setting to incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar. Another aspect of the proposed model is that it remains adoptive to different ontology design patterns without affecting its reasoning capabilities. Our framework views semantic search as an evidential reasoning process, in which we estimate the probability that an entity (e.g. a class instance on the Semantic Web) is relevant to a user's information need, given a query as an initial set of evidence. Moreover, our ranking strategy is grounded on fundamental probabilistic considerations, supported by a well-defined, although densely-structured, sample space. For this reason, the model is simple to understand and establishes a firm ground on which we can evaluate degrees of probability in a very intuitive manner.

3.1.2 Chapter overview

This chapter presents the ground architectural components of the proposed model for semantic search. A hierarchical Bayesian Network is presented that is extracted by means of analysis of the contents and structure of knowledge bases. The resulting network

offers a rich setting for a variety of statistical reasoning capabilities that can satisfy an interesting set of queries. The focus is primarily on assertional semantics, although we do not make such a distinction in the model; our heuristics for evaluating relations are mostly generic constructs of statistical nature, therefore can pertain to any type of semantic relation.

The remaining chapter is structured as follows: Section 3.2 offers a brief overview of Bayesian Inference Networks (or Belief Networks) and their application to IR. Section 3.3 formally introduces our network and its fundamental topological properties. Sections 3.4 & 3.5 introduce the two components of the network, including query construction and modelling by means of a virtual organisation of evidence into layered architectural components.

This chapter and the following (Chapter 4) are strongly correlated. Probability distributions, obtained by considering the frequency, instantiation conditions, and interdependencies of variables in the network, will be presented in full in the following chapter. The ranking strategy and associated inference formulas are also the subject of the forthcoming chapter. The reason we have separated the specification across two chapters is to ease the presentation of the model.

3.2 Bayesian Inference Networks

Bayesian belief networks (Pearl, 1988, 2000; Bishop et al., 2006) are among the best understood stochastic methods for modelling joint probability distributions within a domain of interest. Formally, they are directed acyclic graphs (DAGs) in which nodes represent propositions, or random variables, and arcs portray dependence relations between propositions. Vertices are assigned to every variable in the domain and arrows are drawn toward each vertex X_i from the set of vertices Π_{X_i} perceived to have a direct influence (typically a *causal* influence) on X_i. The strength of these influences are

expressed by conditional probabilities assigned to every variable in link matrix form $p(x_i|\Pi_{X_i})$, also referred to as conditional probability tables (CPTs) in the case of discrete types of networks. These are judgemental estimates encoding our belief that a *child* proposition takes on a value ($X_i = x_i$) given any value combination of its set of *parents*. In principle, the size of a complete matrix specification is exponential to the number of direct parents in the network. For a binary-valued proposition with k parents we must therefore store and estimate a CPT of size 2×2^k. In practise, however, parent relationships are usually structured in prototypical clusters of variables requiring fewer quantifiable estimates, such as Noisy-OR gates (Oniśko et al., 2001; Pearl, 1988). The roots of the network are the nodes without parents and also require a CPT, except it is degenerated into a single row of size n, representing the prior, or marginal probability of the node e.g. $p(x_i)$ for each of its n possible instantiation states.

Conditional probability estimates are consistent if assessed by any set of functions $F_i(x_i, \Pi_{X_i})$ that satisfy

$$\begin{aligned} \sum_{x_i} F_i(x_i, \Pi_{X_i}) &= 1, \\ 0 \;\leq\; F_i(x_i, \Pi_{X_i}) \;&\leq\; 1 \end{aligned} \tag{3.1}$$

where the summation ranges over the states of X_i. The specification becomes a complete and consistent model since the product form $\prod_i F_i(x_i, \Pi_{X_i})$ constitutes a joint probability distribution that supports the dependencies enclosed in the network.

Once factual knowledge about a domain has been compiled into a complete dependency graph, the resulting network becomes a computational architecture for reasoning about that knowledge. The links in the network are treated as message-passing facilities used to propel evidence about the instantiation of variables through the network, allowing us to compute the probability or degree of belief associated with the remaining nodes. Belief propagation is viewed as a generic and sometimes repetitious interaction process between adjacent nodes, which works by looking up values stored in the CPTs

of each intermediate variable. Restrictions on the topology of these networks can lead to different schemes for fusing and combining these probabilities. In general, there are two components that operate independently in a typical propagation or belief-updating process: a top-down form of inference in which parent nodes mediate *predictive* or *prior* support to their children, and a bottom-up evidential reasoning process in which children provide *diagnostic* or *likelihood* support to their parents.

For singly connected networks, it is possible to devise *exact* propagation algorithms to infer the posteriors of all the nodes in a network (reach a state of equilibrium) in time proportional to the network's diameter (Pearl, 1988). The complexity of multi-ply connected networks (networks with cycles) is often treated with approximated or assumption-based reasoning, since propagation with exact algorithms will inevitably fall short (double counting of evidence, loopy propagation), a case generally considered to be NP-Hard (Dagum and Luby, 1993; Cooper, 1990).

3.2.1 Recommended readings for in-depth study

Judea Pearl provides a comprehensive study of Bayesian Networks (Pearl, 1988, 2000). His work is considered the cornerstone of many developments in the field. Chapters 1-4 of Pearl's work on "Probabilistic Reasoning in Intelligent Systems" (Pearl, 1988), with particular emphasis on Chapters 2 and 4, is a highly recommended reading for anyone embarking on a study of Bayesian Networks. Chapter 2 of Pearl's work introduces the basic principles of Bayesian inference and discusses some of the epistemological issues that emerge from the formalism. It is an invaluable reading, particular to readers with no previous exposure to probability theory. We would be doing little justice here had we summarised what is already a very succinct and clear summary of the required concepts to understand Bayesian theory. Chapter 4 goes on to explore a precise propagation heuristic for reasoning in Bayesian Networks. The core of this work has come to be known as Pearl's polytree propagation algorithm (a.k.a. Pearl's *belief* propagation algorithm),

as it is particularly effective and efficient in network architectures that abide to trees and polytrees (Rebane and Pearl, 1987). Pearl's coverage of Bayesian Networks, and particularly his propagation algorithms, is considered a seminal work in the field and one of the earliest introductions of the formalism to Artificial Intelligence.

An interesting summary of Bayesian Networks, with a slightly more general perspective, is provided by Christopher Bishop in his book "Pattern Recognition and Machine Learning" (Bishop et al., 2006). Chapter 8 of Christopher's work deals with graphical models. He offers a concise summary of Bayesian Networks and Markov Random Fields and guides a detailed walkthrough of two general algorithms for inference in graphical models. The algorithms are more general than Pearl's work as they cover propagation in both directed and undirected graphs. This is accomplished via a series of factorisations that convert (or decompose) a graph into a *factor graph* where additional nodes are added to generalise the dependencies in a network. The method presented is also capable of generalising loops in the underlying graph, given that an appropriate factor function is defined. We do not make use of the notation or the more intricate concepts covered by Christopher in our work. Our model and presentations are more inline with Judea Pearl's work.

3.2.2 Relevance to IR

Probabilistic methods in Information Retrieval have been an important instrument for reasoning with uncertainty in a wide range of retrieval tasks. Some of the earliest and pioneering techniques in the field were designed around models that base their core assumptions on rudimental probabilistic and Bayesian principles, such as the binary independence and language modelling approaches (Manning et al., 2008; Croft et al., 2009; Fuhr, 1992). Bayesian Network representations emerged in the late 1980s as extensions of classical probabilistic models and since then have been applied in a variety of ways within the field, both in practical implementations and as conceptual frameworks.

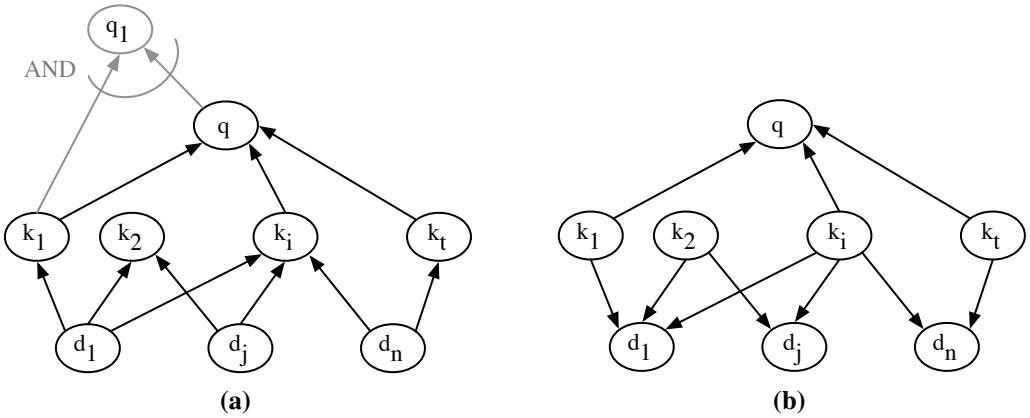


Figure 3.1: (a) The Inference Network model. (b) The Belief Model.

Among the earliest introductions of the formalism to IR have been the works on the *Inference Network Model* (Turtle and Croft, 1991) and *Belief Model* (Ribeiro-Neto and Muntz, 1996). The two models were initially designed as prototypical frameworks aimed to generalise existing approaches (e.g. vectorial ranking) and integrate several sources of knowledge in a single framework (e.g. relevance feedback or multiple document and query representations). In the following paragraphs, we discuss the main properties of the two models as a prelude to the specification of our own Bayesian model in the following section. The two models have been the primary source of motivation for choosing Bayesian Networks as a framework for our model.

A high-level view of the aforementioned models is illustrated in Figures 3.1a & 3.1b. The main difference between the networks is in the directionality of the links. In both models, queries and documents are represented as nodes (q & d resp.) attached to a set of either intermediate nodes (in the case of the Inference Network) or root nodes (in the Belief Model) representing the collection of terms in a corpus. The Inference Network originally organised terms across multiple layers of intermediary nodes. This was used to reflect the potentially multiple representations of terms in a retrieval system e.g. original terms and their stems. The intuition, however, is the same with the more simplified

version in Figure 3.1a. Multiple term representations have not appeared in any inference heuristic investigated in the literature, hence not included in the provided figures. Queries in the two networks can also have multiple representations (q_1 in Figure 3.1a), subject to boolean-like operators (AND/OR) to be combined for potentially improved performance from the same information need.

Nodes in the two networks are treated as binary-valued propositions and are instantiated upon the observation of queries. In both cases, queries are treated as dynamic variables attached to the network upon observation, while documents and terms are static and remain unchanged. Ranking in both cases is orchestrated via a set of inference formulas that reflect the organisation of the graphs. The relevance of a document to a query is interpreted as the probability that a document can materialise given that a query has been observed. From a high level perspective, this carries the same intuition with the *Language Model* presented in the previous chapter (note that the Language Model was introduced later as a formalism in IR).

In the Inference Network, the ranking of a document d_j with respect to a query q is a measure of how much evidential support the instantiation of q provides to the document. Following the directionality of the links, this resolves to measuring:

$$p(q|d_j) \propto \sum_{\forall \overrightarrow{k}} p(q|\overrightarrow{k})\, p(\overrightarrow{k}|d_j)\, p(d_j) \tag{3.2}$$

where $\overrightarrow{k}$ is a vector of terms linked to the query node q. Similarly, in the Belief Model we aim to measure the probability, or *belief*, of a document being relevant, given a query has been observed. This materialises to:

$$p(d_j|q) \propto \sum_{\forall \overrightarrow{k}} p(d_j|\overrightarrow{k})\, p(q|\overrightarrow{k})\, p(\overrightarrow{k}) \tag{3.3}$$

There is a subtle, yet important, difference between the two networks, and hence the outlined likelihood formulas. In the Inference Network, we associate a prior probability

to the document nodes, while in the Belief Model the prior probability is associated to the term nodes. In the latter case, both queries and documents propagate diagnostic evidence to the terms in the network. This results to instantiation of terms that are connected to either the document or the query. In the Inference Network, however, the topology asserts that a document node can only be diagnosed by the terms associated with the query (the diagnostic support flows towards the document, initiated from the query node - in fact, according to the model's formulation, it is not clear what the opposite $p(d_j|q)$ means).

The aforementioned difference renders the Belief Model conceptually more general than the Inference Network (although not necessarily more useful). In the case of a vectorial ranking (Baeza-Yates and Ribeiro-Neto, 1999) (the most common and widely used heuristic for query-dependent ranking), the Belief Model is the only one that can generalise it properly without sacrificing the simplicity of its ranking strategy (inference formula). The denominator of a vectorial ranking resolves to computing the product of the Euclidean vector lengths of two individual vectors: a query vector and a document vector. This includes terms that appear in the query and terms that appear in the document, irrespective if they do not appear in both. With the proper assumptions, the Inference Network can work in a similar fashion, but there will be a sacrifice in simplicity (the inference formula would need to be extended to include propagation from documents back to terms).

Despite their differences, both models are considered pioneering works in the field, providing conceptual frameworks on which additional functionality/information can extend. For more details on their differences, please see (Ribeiro-Neto and Muntz, 1996) and (Baeza-Yates and Ribeiro-Neto, 1999). Indrawan et al. (1994) provide another interesting comparison of the Inference Network from the perspective of an alternative implementation.

Later works in the literature have extended these ideas to incorporate additional

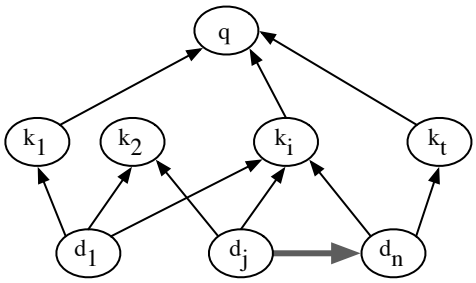


Figure 3.2: The Inference Network model incorporating a document dependency.

features into the ranking process, such as document structure (Myaeng et al., 1998; Crestani et al., 2003) and hypertext link analysis (Croft and Turtle, 1989; Calado et al., 2003). For example, Croft and Turtle (1989) extended the Inference Network with additional dependencies between document nodes to reflect the existence of links between hypertext documents. This is illustrated in the striped-down version of the network in Figure 3.2. In this example, an additional link has been added between nodes d_n and d_j. This potentially reflects a hypertext link (or set of links) from a hypertext document represented by d_j to a document represented by d_n. In terms of inference, this can translate to the introduction of probabilities of the form $p(d_n|d_j)$ in the expression:

$$p(k_t|d_j) = p(k_t|d_n)\ p(d_n|d_j) \tag{3.4}$$

Intuitively, if hypertext node d_n is indexed by a particular term k_t and is linked to node d_j, then there is some probability that node d_j should also be indexed by that term. How the dependency is quantified can be subject to both the popularity and the number of links connecting the two documents (the authors have left this abstract). The inclusion of such dependencies between document nodes creates additional loops in the network, although their treatment in the inference is again subject to interpretation. For example, in a strictly bottom-up propagation process, we avoid re-counting the same evidence in the evaluation of d_n (since there is no notion of predictive/top-down support). Note that

the definition of the dependency portrays a form of diagnostic support from node d_n to d_j. The opposite (prior/predictive support to d_n from d_j) is not clear what it may mean. However, reasoning in the Inference Network is largely defined as a means of diagnosing documents in response to a query.

Successful implementations of Bayesian Networks (not related to the above models) are also found in document clustering and classification (Denoyer and Gallinari, 2004), conversational agents (Kim et al., 2007), and other related fields. Precise propagation and reasoning in Bayesian IR networks remained intractable tasks, and their design was largely focused on the interpretation of complex dependencies as canonical functions that are practical and easier to implement. The probabilistic functions outlined in the inference formulas above are generic functions that can be interpreted as any form of frequency measure, as long as the underlying probabilistic foundations are not violated.

Our work has been largely motivated by the Inference Network. We find the topology of the network attractive for the following reasons: (1) it provides a clear solution for diagnosing resources from evidence in the query, and (2) the position of documents as roots in the network enables them to be potentially predicted or influenced via support from other sources of evidence. The clear separation of documents and queries (they become independent once term nodes are observed - see the *2nd* rule of conditional independence in (Bishop et al., 2006, page 375)) and the position of documents as roots in the network, allow us to easily extend the framework with links between document nodes to model the interrelation of resources via object properties on the Semantic Web. As will be covered in the following sections, we will model such additional links between entity resources (document nodes in the above works) to propagate the relevance of entities to other entities across the network.

3.3 Model Overview

The Bayesian inference model designed for the task of entity search is illustrated in perspective view in Figure 3.3. Example networks on fictitious data are presented in Figures 3.7 & 3.6 (we will come back to these later in the presentation). From the outset, the model consists of two component networks: a static *resource network* containing information about data resources and their semantic interrelations, and a dynamic *query network* containing a (tacit) specification of the user's information need. The model differs from related works in that it aims to represent and make explicit in the inference the presence of multiple object and datatype relations that may potentially bind or link semantic resources together and with primitive data values, as it is customary with Semantic Web data. Furthermore, as will be explained shortly, the model exposes additional functionalities to the inference process. Part of our goal in designing this model has been to enable reasoning with more complex or expressive information needs, with semantics specified explicitly by users or incorporated via more implicit bindings. In summary, fixing the instantiation of resources based on evidence in the query will affect the flow of propagation via associated dependency links, hence allowing query semantics (mostly of implicit nature) to affect the inference.

The *resource network* is a dense hierarchical network intended to capture and quantify both assertional and terminological semantics as probabilistic dependencies among binary random variables. The network is built once for a given collection and remains unchanged during query processing. The query network is a dynamic component in the architecture represented by a single leaf node and a set of two distinct virtual sub networks/components. The *query network* is a temporal network created whenever a user queries the collection and only exists during query processing. Once a result is obtained, the query network is discarded (unless further processing or expansion is expected). A query encloses the initial evidence to be attached to the resource net, and we explore two such types, as indicated in the diagram. Nodes in the resource network are binary-valued

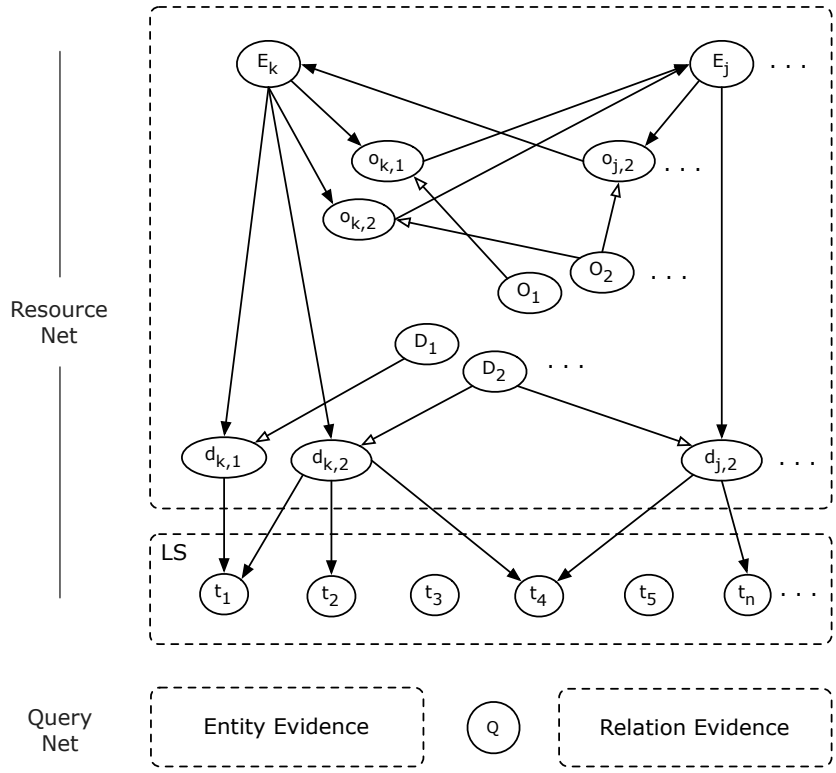


Figure 3.3: Perspective view of the model.

propositions and can take on values from the set $\{true, false\}$. Query nodes are always assigned the value of *true*, indicating that an information need has been observed and the corresponding query formulated.

Mappings between the two networks will determine the inference paths to be traversed in order to evaluate entity resources against the query. The Literal Space, marked as LS in Figure 3.3 acts as the main mapping facility between the query and the rest of the network. The LS contains an assortment of text representation nodes extracted from the primitive data-type values in the knowledge base (more info on the LS in the respective section). Mappings are dynamic and can entail topological restrictions on the inference process and instantiation entailments of resource variables. We will detail and provide examples of the process and each respective component in the forthcoming sections.

Being a fully engaged Bayesian Network, the model can, in principle, support the evaluation of almost any vertex in the network, given some set of initial instantiation evidence. As it is customary with Bayesian Networks in IR, we will treat the model as an expressive architectural framework on which we can approximate reasoning using various generic functions of standard IR schemata (e.g. functions to estimate term frequency, field weighting, and link proximity). Our focus is on the evaluation of entity resources, marked as $E_{...}$ in the model. All of our assumptions will be defined accordingly to reason with entity variables. The presence of cycles and a potentially large sample space precludes the development of exact recursive inference as originally conceived for Bayesian Networks. The complexity of multiply connected networks is often treated with approximated or assumption-based reasoning, since propagation with exact algorithms will inevitably run into trouble (double counting of evidence, loopy propagation), a case generally considered to be NP-Hard, both in the exact and the approximate case (Dagum and Luby, 1993; Cooper, 1990). We will use various approximations/generalisations and prototypical forms of interaction (e.g. Noisy-OR gates (Oniśko et al., 2001; Pearl, 1988)) to isolate parts of the network and reason with in a timely manner for the task at hand. On the whole, retrieval will be geared in terms of the concurrence of two estimates: *entity-diagnosis* and *entity-prediction*. How these are extracted and coordinated will be the subject of this and the forthcoming chapters.

3.4 The Resource Network

The underlying building block of semantic knowledge bases is a subject-predicate-object triple, whereby subjects and objects are allowed to be interchanged. A knowledge base may be thought of otherwise as a loosely coupled directed labeled graph (DLG), where subjects and objects are treated as nodes and predicates as labeled edges (relations) between them. DLGs are a common and generic model to describe possibly any type of

semantic network or association graph. On the Semantic Web, relationships are first-class URI resources and can be defined locally or reused from existing vocabularies. The goal of our translation is to devise a generative model for projecting the DLG manifestation of knowledge bases to a form of DAG, on which we can delicate retrieval of resources to an evidential reasoning process. The outcome is not initially acyclic, cycles exist in the model, but this is a common scenario with Belief Networks and will demand special treatment and reasoning during the inference process. The resulting model is not necessarily restricted to Semantic Web data, since a translation from a DLG model can have a broader perspective. Dependence implications from assertional and terminological semantics (e.g. subsumption and inclusion information) that are common constructs in semantic networks will be treated by the same general-purpose statistical schemes. The precise translation choices and topological properties of the two networks are defined in their respective forthcoming sections. The following terminology will remain fixed, although with arbitrary content:

- $\mathcal{U}$ is the set of all resources in a knowledge base that participate in a subject-predicate-object triple
- $\mathcal{S} \subseteq \mathcal{U}$ is the set of all subjects
- $\mathcal{O} \subseteq \mathcal{U}$ is the set of all objects
- $\mathcal{L} \subseteq \mathcal{O}$ is the set of all literals or primitive data-value objects
- $\mathcal{R} \subseteq \mathcal{U}$ is the set of all properties/relations

Subjects and objects are allowed to be interchanged, hence the condition $\mathcal{S} \wedge \mathcal{O} \neq \emptyset$ can hold, given the completeness of the working set. Relations are partitioned into object properties $R_O \subseteq \mathcal{R}$ (linking resources together) and datatype properties $R_D \subseteq \mathcal{R}$ (linking resources in $\mathcal{S}$ to literals in $\mathcal{L}$). The subsumption $\mathcal{R} \subseteq \mathcal{S} \cup \mathcal{O}$ is also true, since a property can itself be the subject or object of a different relation.

There are two main types of nodes in the resource network: nodes depicting candidate entities for retrieval (we will refer to them as *entity members* or *member variables*) and

nodes depicting relations between entities and with primitive datatype values (otherwise, object and datatype property nodes). Property nodes are demarcated between local and global, as will be explained in detail shortly. A local context for each entity is defined in the model, reflecting the local use of semantics in the model (datatype and object relations). Arcs between nodes define probabilistic dependencies and act as evidence passing and filtering facilities. In neural science, these would be equivalent to synaptic connections between neurones.

3.4.1 Entity members

A subset $E \in \mathcal{S}$ from the knowledge base is selected as candidate for retrieval and translated to n binary random variables, $\{E_i, ..., E_n\}$ in the Bayesian Net. We keep the definition of E arbitrary for now and include any one or more first-class resources that participate in a triple. A member variable set to true ($E_i = true$) is said to be activated by the query for evaluation (according to our earlier definition, this may include either relations and/or subjects). Activation of member variables is subject to whether a diagnostic (bottom-up) path is open between the member variable and evidence in the query. A path is initiated via a mapping to the Literal Space through which diagnosis can reach the member via any number of *datatype* properties (covered next). Query evidence will arrive at various locations from the Literal Space and propagate in a bottom-up fashion towards the member variables (either directly from the Literal Space or across several other intermediary members). Entity members are evaluated in isolation, so each will consume a separate propagation process. Details on the inference process will be covered in later chapters.

Figure 3.4 shows the two paths through which diagnosis can reach member variables. A path between a candidate entity and the LS may run through

1. the member's *local datatype context* (Figure 3.4a), which includes a set of local datatype properties (e.g. $d_{i,j}$ for member E_i), and

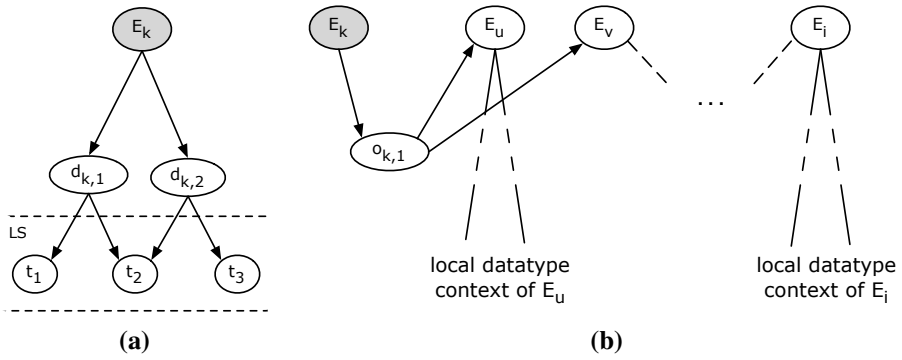


Figure 3.4: Diagnosis reaching a member variable via (a) the member's local datatype context, and (b) the local datatype contexts of other entities.

2. through the local datatype contexts of neighbouring and/or distant members linked via object properties (e.g. $o_{i,j}$ for member E_i) or a chain of properties via any number of intermediary members (Figure 3.4b).

The latter constitutes the *local object context* of entities (or a chain of such contexts across intermediary variables). The entire network is thus decoupled between the local datatype and local object contexts of entities. A member variable will be set to $true$ only when any of these paths contains a binding to query evidence. A binding to query evidence will mostly involve instantiation of nodes in the LS, although other restrictions are applicable (e.g. the strength of instantiation of local property nodes). In any other case the variable remains in a $false$ state. Consequently, retrieval considers entity members that have been activated as $true$ and will dedicate a separate trial for each. Active members are either treated for evaluation or used to support the evaluation of other entities. The inference process will be the focus of our ranking strategy and will be presented in the following chapter (Chapter 4).

Candidate entities are portrayed as dependent on the local object properties of other entities (e.g. $p(E_k|o_{j,2})$). We will cover property nodes next. This type of dependency is the result of backlinks, which are strictly used to solidify the conditional interdepen-

dence of entities through which diagnosis from the LS can reach members via connected resources (as explained previously). Backlinks are *not* used to model or quantify predictive support for entities at this point. This dependency will be reduced to a (constant) prior probability $p(E_i)$ later to keep a working model in order and will form our first level of approximation or assumption. In a later chapter, we will present possibilities for extending the basic model via explicit use of backlinks and dataset information to simulate an additional link analysis layer in the inference. For now, we do not explore backlinks or dataset information, hence the network is clamped at each member.

3.4.2 Property nodes

The set of properties $\mathcal{R}$ in a knowledge base is composed of two different sets[2], $\mathcal{R} = R_o \cup R_d$: The set $R_o = \{O_i, ..., O_n\}$, containing binary random variables representing the n translated object properties from the knowledge base, and the set $R_d = \{D_i, ..., D_n\}$, representing the n translated datatype properties. Property nodes in the Bayesian Net are separated between *local property nodes* (local to each entity member) and *global property nodes* (global across the entire knowledge base). The aforementioned definitions correspond to global property variables. The reason for defining two types of properties is pragmatic and will be explained shortly.

3.4.2.1 Global property nodes

Global property nodes are dependent on term nodes in the LS representing the actual labels associated with properties in the knowledge base. Consider Figure 3.5 for a finer example of property nodes with labels projected as term nodes (indexes) in the LS. In our current implementation, labels are extracted directly via *rdfs:label* relations appearing in the property definitions (in the case of Semantic Web data) or deduced from the property

[2]We will use the notation O_i (D_j respectively) to refer to the actual datatype and object properties and also their translated variable nodes in the Bayesian Net.

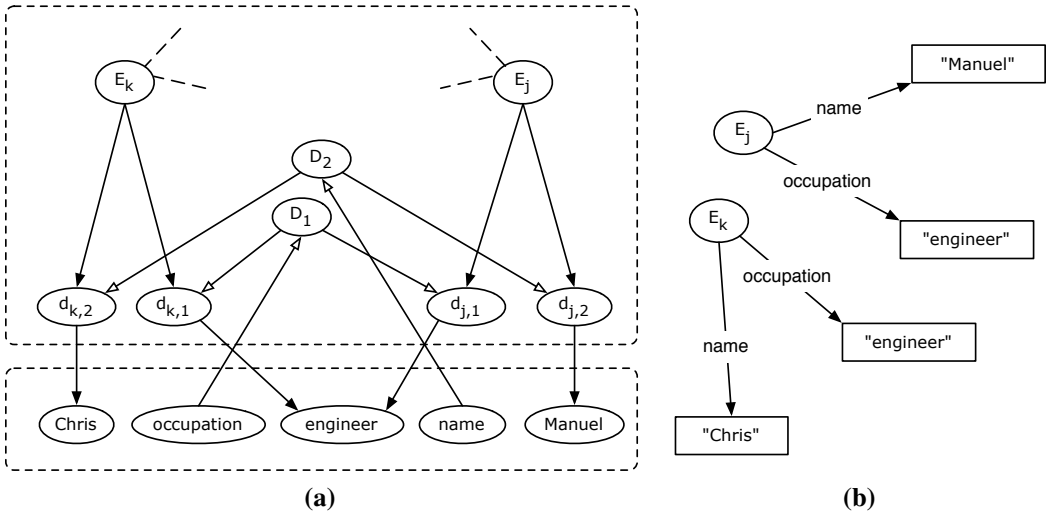


Figure 3.5: Example of global datatype property nodes (D_1 and D_2) with their labels projected as indexes in the Literal Space. Global object property nodes are treated analogously. Figure 3.5b contains the original DLG.

URIs via simple heuristics (e.g. where *http://xmlns.com/foaf/0.1/name* resolves to *name*) when no such label exists. Other label forms may be preferred over *rdfs:label* during the translation phase. Details of implementation are irrelevant at this point, but for the experiments in Chapter 6 we have used the above method with additional processing of the extracted terms. Property labels are used to establish mappings with the query network, allowing properties to be diagnosed as potential query elements.

A global property node set to $true$ ($\{D_i, O_i\} = true$) is said to be instantiated by evidence in the query. This type of evidence is predictive, as indicated in Figure 3.5, hence the prediction of properties entails their instantiation state. Observed properties will be used as *logical conditions* to delimit the instantiation of local properties as a result of the mapping to the query. Instantiation of global properties will not be fused in the inference process but will be used to render/influence the dependencies associated with local property nodes and the candidate entities. Hence, this is a model for enabling query semantics to influence the diagnosis arriving at entities. For the same reason we refrain

from complicating the general network in Figure 3.3 with explicit links to the Literal Space. The precise conditions for the instantiation of global property nodes are interlined with the treatment of property evidence in the query and are presented later in Section 3.5.

3.4.2.2 Local property nodes

Local property nodes ($d_{i,j}$ or $o_{i,j}$ in Figure 3.3) are defined to associate higher order properties (global properties) to a local context defined by each individual entity member. Local properties descend from a single global property node and a single entity member, which act as the parents of the respective node. The naming convention used to distinguish properties is adopted to reflect its parents e.g. a local datatype property with parents E_i and D_j is named $d_{i,j}$ accordingly. Local property nodes are conditionally independent with each other, given their set of parents and children in the Literal Space. There are no direct connections between them and can have multiple descendants in the network e.g. they can link to several entity members in the network (case of object relations) and to several term nodes in the Literal Space (to connect the literal indexes to the member variables).

A binary value ($true/false$) associated with a local property reflects the instantiation of the corresponding global property node i.e. a variable is set to $true$ exactly when its parent property is $true$ ($o_{i,j} = true : O_j = true$).

$$o_{i,j} = \begin{cases} true & \text{if } O_j = true; \\ false & \text{otherwise.} \end{cases} \tag{3.5}$$

Consequently, properties will be marked as either $true$ or $false$ exclusively in each query evaluation. The conditional probabilities associated with local properties (either $p(d_{i,j}|E_i)$ or $p(o_{i,j}|E_i)$) based on their states are the main methods for external parameterisation of the model, and quantities will vary according to the type of query formalism explored

(whether evidence should be treated as more explicit as opposed to more implicit).

Local property nodes have a significant role in the network. First, they facilitate the translation of bidirectional use of properties on the Semantic Web, something not possible with a global form of property representation alone. Recall that in a triple-based knowledge base the same relation can be used to link to and from the same entities. In other words, entity x can be the subject of triple with predicate p and object y (as in $x - p - y$), and the object of another triple with the same predicate z and subject y (as in $y - p - x$). Such two triples define a form of bidirectional association between two nodes with the same relation (even though it may appear as a different link in the DLG - the label would be the same). In Bayesian Networks links are directed and a node can only exist on one side of the relation i.e. nodes cannot be both the cause and effect in a given relation. Therefore, there needs to be some form of intermediate node to define the type of association/predicate.

Second, local properties delineate a clearly defined sample space on which paths from the Literal Space can be quantified according to the local context of individual entity members. For example, $p(t_k|d_{i,j})$ allows quantifying the relation of term node t_k to a specific datatype property in the context of E_i (essentially via the property represented by D_j). This brings together an interesting formalism for traditional weighted-field retrieval, essentially treating an entity member as a structured document, but with the added expressivity due to the different instantiation states of properties. The latter allows interpreting evidence of relations in the query in the evaluation process, which is a desirable property for resolving more complex queries.

3.4.3 The Literal Space

The set of all literals $\mathcal{L}$ in a knowledge base is enclosed within the LS at the lowest level of the network in Figure 3.3. Every node in the LS corresponds to an index term extracted via some form of term extraction technique. For example, if the string

"semantic search" has been extracted into the distinct terms "semantic" and "search", then two representation nodes are created. We treat the LS as a single layer containing one form of indexes per collection. This may include, for instance, the stems or the original form of text in the knowledge base. The precise term-extraction techniques applied during evaluation of the model will be covered in Chapter 6.

The set $U \subseteq \mathcal{L} : U = \{t_1, ..., t_n\}$ represents the set of all index terms extracted from a knowledge base (including property labels), modelled as n random variables. Term nodes are considered conditionally independent with each other given their set of parents (local datatype properties) and children (global property nodes). There can be several paths between the nodes in the LS and the local contexts of entities (e.g. a literal associated with *foaf:name*, *dc:title*, and *rdfs:label* bounded to the same entity), and terms can be shared across member contexts. The dependency of global properties on term nodes (as illustrated in Figure 3.5) asserts that prediction of properties will be initiated from inside the LS, although the connection is treated like a decision link, since predictive evidence will not be fused further in the network.

The LS exposes a natural interface between the query and the rest of the network. Evidence will initially flow from the query network to the Literal Space and propagate through the rest of the network by unfolding the space covered by term nodes, for every entity member being evaluated. A binary value ($true/false$) associated with a term node indicates whether the term is observed by evidence in the query. Query evidence need only attach to the Literal Space, while different propagation signals using different combinations of query nodes can result in a variety of expressive query formalisms. A term node set to true ($t_i = true$) is said to be instantiated by evidence via a mapping to the query.

3.5 The Query Network

The query network is a *virtual* component in the architecture and reflects the overall strategy for meeting a user's information need. In general, we treat information requests as tacit specifications of a data resource, provided as either a combination of keywords or a form of semi-structured natural language description, which remain mostly ambiguous and internal to the requestor. This may include, for example, a full-bodied description for which a user is seeking artefacts and objects to link to or a typical keyword search over a semantic blog, wiki, CMS, or any other form of knowledge base front-end. A ranking strategy is intended to transform these implicit specifications into an execution plan for evaluating and retrieving instances from a knowledge base.

Query evidence is enclosed within two distinct layers: Entity Evidence and Property Evidence. Query layers depict different aspects of a request (e.g. the presence of a literal or a property definition) and are evaluated in combination for potentially more optimal results. We expect that queries of the form "*person* ***named*** *Jim Smith*" or "***friends of*** *Jim Smith*" will be treated with special emphasis on their semantics (the datatype relation "*named*" and object relation "*friends of*"). It will be possible to evaluate several such patterns in a single query e.g. "***friends of*** *Jim Smith who* ***live in*** *California*". The semantics are implicit and should not block any other paths in the model. Part of the evaluation (Chapter 6) will involve determining the context in which the type of query representation is most effective and the proper degree of influence these semantics should have on the retrieval process. Ideally we would want to maximise precision without affecting recall in the final results.

The contents of the layers are the initial evidence to be transmitted and factored into the resource network. As a bare minimum functionality, our focus is on query layers that are induced via fully automatic means. Query layers are treated as very ambiguous specifications of the aspects they intend to cover, thus their impact remains implicit, just enough to intensify the probability of observing the corresponding resources in the query.

Manual query construction can aid to transform evidence into more explicit provisions for the inference process, thus facilitate better understanding of the user's intent.

Ad-hoc processing of queries aims to extract knowledge from ambiguous information requests without requiring extensive user involvement. A description, set of keywords, or other tacit specification is provided by the user, and the query engine takes care of the mappings to the resource network. A pre-processing stage may involve linguistic and syntactic analysis to improve the results of the mapping process. For the moment, it is assumed that there is utmost one-to-one mapping between query terms in the respective layers and term nodes in the LS. Figures 3.7 and 3.6 illustrate two example queries laid out across the two layers. Query layers attach to the LS by a set of unquantified (dummy) links, and their purpose is to instantiate term nodes to some initial state. Hence information flows one way only − from the query layers to the variables affected by the observations. The query, in effect, instantiates a part of the network composed of the nodes and links participating in the computation. The contents of each layer are explained next.

3.5.1 Entity evidence

The first layer, q_1, encloses a set of independent dummy variables representing the (processed) terms in the user's query that match to indexes descending from local datatype properties. This excludes terms associated with global properties nodes. Every node in this layer is considered a disparate frame of knowledge that will be used to propagate diagnosis to the *local datatype contexts* of entity members. Nodes that do not match to index terms contain no mapping to the LS.

3.5.2 Property evidence

The second layer, q_2, encloses a set of potential property definitions present in the query. Nodes in this layer attach to terms in the Literal Space linked to global property nodes. The idea is that a strong evidence in the query may instantiate a global property node to $true$. Since global properties influence directly the instantiation of local property nodes, this can intensify or weaken the evidence that flows through the local context of entity members (initiated from q_1) via the respective local property node. This, in turn, will solidify in the inference the presence of a relation in the query. The strength of evidence that flows through local property nodes is weighted on the conditional probabilities associated with local property nodes given the entity that they descend from (e.g. $p(o_{i,j}|E_i)$). Conditional probability assignments are covered in the following chapters.

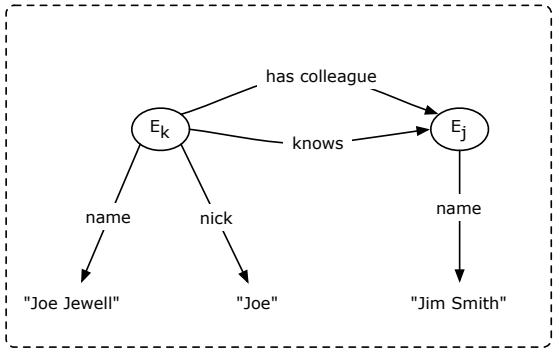
In the case that global properties are associated with several terms in the LS, then we must decide whether there is enough evidence in the query to affect their instantiation. Assuming θ denotes the index terms $t_i, ..., t_k$ in the LS associated with a global property node O_i, such that the specification $p(O_i|t_i, ..., t_k)$ is satisfied in the model, then O_i may be instantiated according to the following condition:

$$O_i = \begin{cases} true & \text{if } p(q_2|\theta) > \gamma : \gamma \in [0, 1); \\ false & \text{otherwise.} \end{cases} \tag{3.6}$$

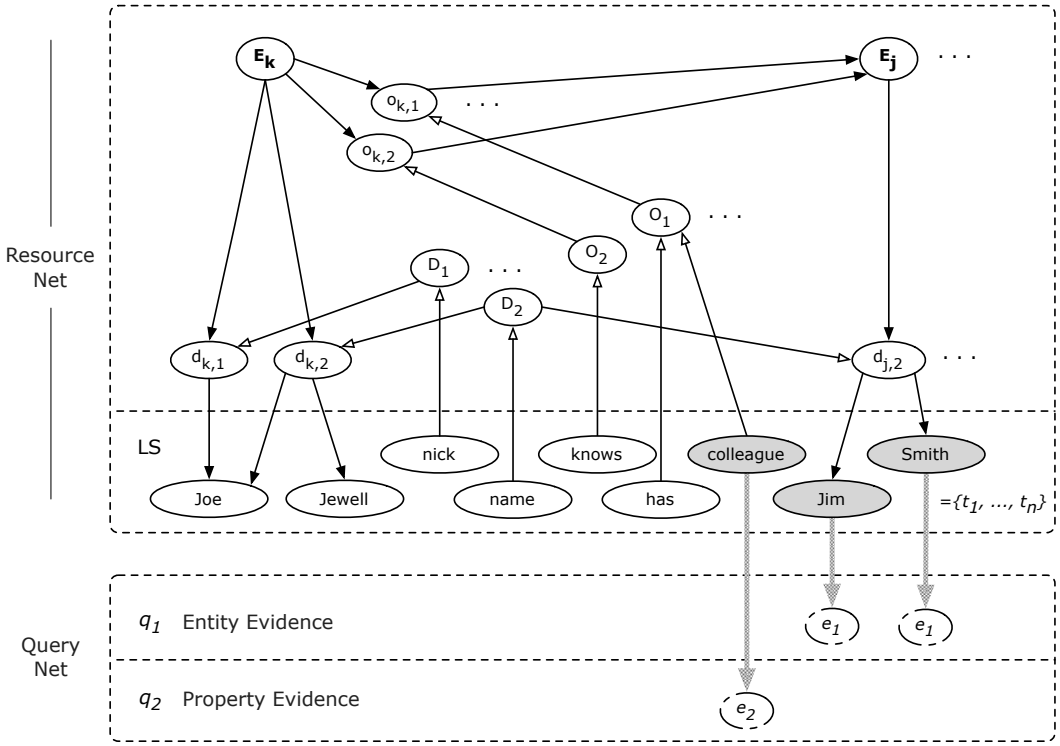
$$\text{where} \quad p(q_2|\theta) = \frac{p(q_2, \theta)}{p(\theta)} \tag{3.7}$$

We may treat $p(q_2, \theta)$ in prototypical form as the intersection of terms associated with variable O_i and the ones present in the query layer q_2. $p(\theta)$ is treated analogously over terms associated with the property O_i alone. If we exclude any other links to/from the corresponding indexes in the LS, then Equation 3.7 is a viable approximation of the

degree of coverage of the property definition (associated indexes denoted by θ) by the respective query layer (q_2). We would want to keep the estimation simple and efficient, since this is a pre-processing step in the evaluation. The threshold γ can be fiddled during implementation of the model. Verbose property labels should require a higher threshold. For single-term labels the parameter can be set to 0 (hence any single mapping should suffice). For a finer treatment, we can enumerate the properties in a knowledge base and issue queries involving the property labels alone and observe how the system reacts or manages to establish the correct mappings.



(a) DLG



(b) Translated BN

Figure 3.6: (a) Original DLG data fragment. (b) Translated Bayesian Net with example query network for a request for "colleagues of Jim Smith". The two query layers are treated separately with q_1 instantiating nodes to participate in propagation and q_2 instantiating nodes to influence the states of global properties.

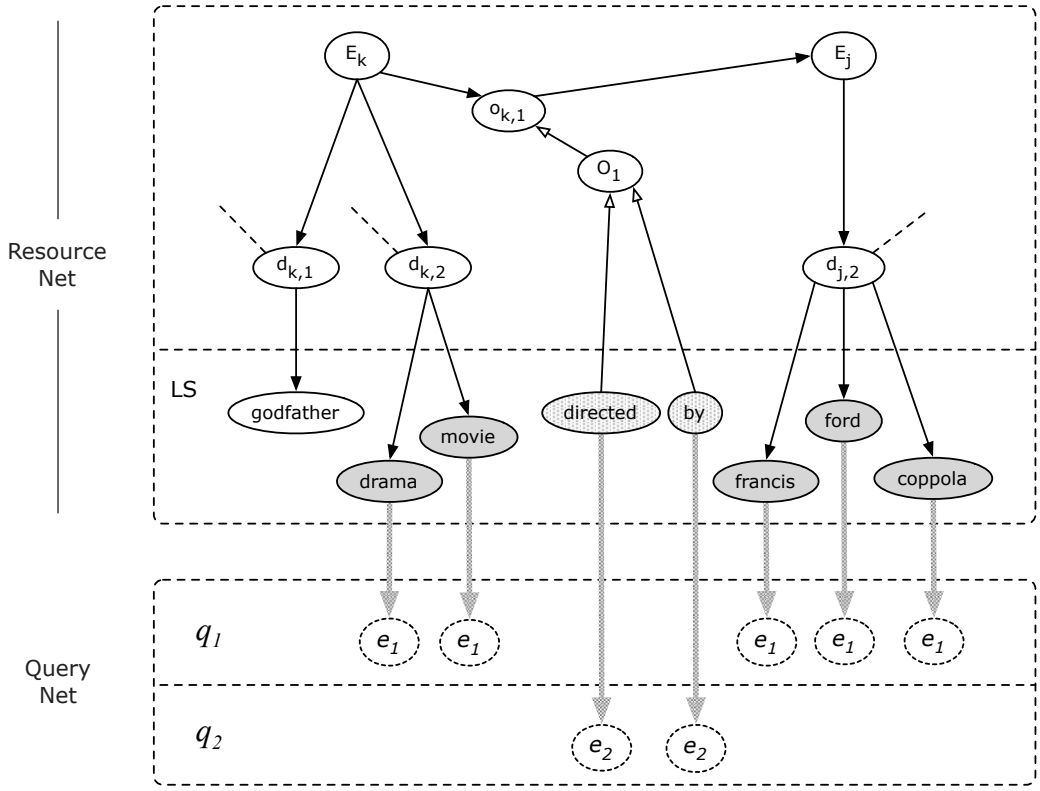


Figure 3.7: Example query network for a request for "drama movies directed by Francis Ford Coppola". The two query layers are treated separately with q_1 instantiating nodes to participate in propagation and q_2 instantiating nodes to influence the states of global properties.

Chapter 4

Completing the Model: Probability Estimates and Inference

4.1 Introduction

In the previous chapter, we have presented the topological properties of a Bayesian inference model to support searching of entities in Semantic Web graphs using free-form or semi-structured natural language queries. The resulting network was the outcome of a generic translation from a directed labelled graph (DLG) data model, a common representation formalism for semantic and association-based knowledge bases. In this chapter, we unify the model into a complete specification by presenting conditional probability assignments and a unified ranking strategy for inferring the impact of evidence when a query is executed. The chapter is divided into two subsections respectively. In the forthcoming chapter, we will revisit all of the instantiation conditions discussed in this and the previous chapter to guide an evaluation and analysis of the model over a realistic data collection.

In case of ambiguity, please keep a closer look at one of the example diagrams presented in the final section of Chapter 3.

4.2 Estimating Conditional Probabilities

In order to complete the translation and firm up the model for inference, the remaining issue is to quantify the conditional and marginal probabilities for all the nodes in the network. The resulting distributions will be unified and organised into inference formulas that will form our ranking strategy. Conditional probabilities are the mechanisms by which we reason in the model, in essence giving us a quantitative perspective over the dependencies in the model. Estimates are required for four different node types: term nodes in the Literal Space (LS), local object and datatype property nodes, and entity members. Global property nodes have already been treated in the previous chapter.

Some clarifications before moving on

The arbitrary complexity and size of the model suggest that we must seek alternative strategies, beyond exact heuristics depended on precise Conditional Probability Table (CPT) specifications, if we are to achieve computationally tractable inference in the network. Associating with every variable a CPT that enlists probability estimates for all possible value combinations of its parents is rather impractical, if at all feasible, since the construction of exact CPTs requires prior knowledge of the type and number of parents being conditioned. Several nodes in our network, however, can be related to an arbitrary number of parents, since the network is the result of a translation from an arbitrary knowledge base. It is also common that, in many practical situations, interactions among propositions need not be handled by listing all possible combinations of conditional probabilities, but by manipulating only sets of propositions and their states.

In this work, we attempt to interpret complex dependencies as canonical functions that are practical and easier to implement. The complexity of the model will be further simplified by restricting propagation to delivery of evidence to specific nodes in the network. In this way, we can deal with propagation in a more deterministic and less

ambiguous manner. At the same time, we try to stick to a model that is closely tied to its formal probabilistic underpinnings in as much as possible. The ground foundations of the model offer a rich setting to incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar. Many of our probability assignment choices are also tightly coupled with the assumptions in the final ranking strategy. For example, many complex interactions/dependencies will be enclosed within prototypical functions that resemble traditional scalar and other functions used in IR (hence can affect how the estimates are computed).

4.2.1 Term nodes

Instantiation of term nodes is determined by mappings to the query network, which attaches to the LS for propagation. Term node dependencies are defined by conditional probabilities of the form $p(t_v|d_{i,1}, ..., d_{i,n})$, assuming a term t_v and a set of n local datatype properties associated with class member E_i. In quantifying the dependency of term nodes on local datatype properties we make the following assumptions:

Assumption 1: The presence of a term in the value associated with a datatype property (portrayed as $p(t_i|d_{i,j})$) is independent of the term's relation to other properties associated with the same entity.

Assumption 2: The presence of a term in the value associated with a datatype property or the label of a global property is independent of the presence of other terms.

Assumption 2 is a typical term independence assumption. Assumption 1 is crucial for completing the specification of term dependencies without requiring exhaustive enumeration of detailed interactions with property variables. Assumption 1 has a direct implication on the inference process. By allowing properties to be treated independently

we may treat the likelihood observed from a term's association with a particular property as a sufficient condition for diagnosing an entity member from the query, regardless of the instantiation of other properties.

Given the above assumptions, we can decompose the specification $p(t_v|d_{i,1}, ..., d_{i,n})$ into a series of prototypical functions $f : (t_v, d_{i,j}) \rightarrow W_{d,t}$ over all property nodes ascending from t_v. In effect, if we consider Πt_v to be the parents of term t_v the weighting function can provide a value $p(t_v|d)$ for every $d \in \Pi t_v$. $W_{d,t}$ can feature the effect of an indexing weight, such as the relative frequency of a term inside the value associated with the respective property. The function does not discriminate over the states of the parent variable, since there is practically little reason to interpret $p(t_i|d_{i,k} = true)$ and $p(t_i|d_{i,k} = false)$ differently. Furthermore, we do not quantify the false states of terms ($t = false$) since only terms present in the query engage in propagation. Term weights need not be probabilistic estimates at this point and can feature any variation of a TF-IDF weighting scheme (Salton and Buckley, 1988), as long as evidence is normalised before reaching the entity variables. During inferencing (Section 4.3), the values are normalised at the property node via a cosine rule that ensures evidence propagated further up ranges between $[0, 1]$. In the following chapter, we evaluate an implementation of the model using logarithmic frequencies for more efficient indexing.

4.2.2 Local object and datatype property nodes

The bindings of local properties, object and datatype, to entity variables enable the use of additional filters on the diagnostic evidence accumulated at entities. These are portrayed and quantified by conditional probabilities of the form $p(d_{i,j}|E_i)$ and $p(o_{i,j}|E_i)$. The relations are always one-to-one since local property nodes are local to each entity member. The values associated with the dependencies will be used to fine-tune the strength by which diagnosis (bottom-up propagation) arrives at entities. Since properties are marked as either $true$ or $false$ in each query evaluation, the values are strictly

differentiated on the instantiation of property nodes. Recall that local property nodes inherit the instantiations of global property nodes, which are instantiated according to their mappings to query evidence.

For a given local property node $d_{i,j}$ its conditional probability on entity E_i is assigned as follows:

$$\begin{gathered} p(d_{i,j} = true|E_i) = q, \\ p(d_{i,j} = false|E_i) = 1 - q, \\ where \;\; q = [.5, 1] \end{gathered} \tag{4.1}$$

Higher values for q are expected to intensify the presence of the respective properties in the query. The intuition is that by tweaking q to a higher interval, the evidence in the query becomes more explicit when traveling along the weighted path. The extreme case, i.e. setting $q = 1$, will block any evidence traveling along the axis $t \longleftarrow d_{i,j} \longleftarrow E_i$ from reaching the member variable when $d_{i,j} = false$ (property D_j not diagnosed in the query), but will propel diagnosis when $d_{i,j} = true$. However, using the extreme case is not functional, since all other paths will be blocked. The idea is that we want to allow expressive query patterns to be evaluated on the same canonical framework as simple keyword queries, without necessarily requiring interference to the process by the user (e.g. selecting whether expressive or simple patterns are to be executed).

The above formalism provides a mechanism by which we can treat the evidence of a property in a query as a more explicit provision in the inference process. The idea is that when we want to intensify the presence of a relation in a query (such as in a query for "datasets **released in** 2010"), we can increase the value of q, which will allow evidence to have a stronger impact when it flows through that relation (represented by the dependency link). Setting q to 0.5 would nullify the effect of instantiating a property to $true$ or $false$, since the strength of the conditional dependency would be the same in both cases (hence evidence flowing through "released in" or any other property would

have the same impact).

4.2.3 Entity nodes

The conditional probabilities of the type $p(E_i|o_{j,k})$ are used to portray a form of proximity measure between entities in the context of the relation being considered. Thus, the impact of diagnosis reaching a member variable from the context of another variable will be adjusted by a form of semantic similarity w.r.t. a particular object property. The distributions will be treated in canonical form; therefore, as before, exhaustive enumeration of parent values will not be necessary. During inferencing, multiple connections between entities will be treated independently by encoding them into a disjunction operator.

Our criteria for estimating $p(E_i|o_{j,k})$ is to assign a value based on the authority of E_i (the entity propagating its diagnosis) in the context defined by the property represented by $o_{j,k}$. We wish to differentiate against very common properties such that uncommon events are more profound in the inference, which are expected to have a special meaning to the entities involved in the relation. In effect, the more unlikely a relation, the stronger will be the evidence, considering the popularity of the given entity in that type of relation. We extrapolate the probability over the global property nodes, thus being able to simulate an estimate over the entire network. By rule of conditional probabilities we define:

$$p(E_i|o_{j,k}) = \frac{p(E_i \wedge O_k)}{p(O_k)} \tag{4.2}$$

where $p(E_i \wedge O_k)$ may be interpreted as the number of entities that use O_k to link to E_j (backlinks pertaining to O_k) and $p(O_k)$ may be defined as either the number of entities that use O_k (number of $o_{\_,k}$ local properties in the network) or the number of times the property represented by O_k is used in general (number of total backlinks pertaining to O_k). The latter will give emphasis to how frequently the relation appears within the local contexts of entities e.g. if everyone *knows* person x, it does not necessarily make him/her

popular if they all know many other people − so a value such as $2/2$ may become $2/2000$ even though only two entities define the relation locally and both link to x. The inverse of the equation would be similar to a conventional IDF metric, except we are looking at a particular entity and a particular type of relation.

4.3 Ranking Strategy

A ranking strategy outlines the inference formulas that will consume propagation in the model and infer the impact of query evidence on candidate entities. Evaluation of entities is performed by using statistical inference to propagate belief values across the network and retrieving the members that rank highest on their posterior estimation.

Our intuition for ranking is that every entity member is treated separately for evaluation. The knowledge base, therefore, gets partitioned between two, possibly uneven, disjoint parts in every evaluation: events that relate to the given entity and events that do not. When a query is issued to the system, it is treated as an observable event that is intersected with the partitions of the universe. What we are interested to measure is the degree of coverage by the query of the space covered by a given entity. Considering an entity E_i and a query specification Q, our goal is to estimate $p(E_i|Q)$. Intuitively, this accounts to the probability that an entity is retrieved (i.e. the belief we accord to hypothesis E_i), given that a query has been observed (i.e. given that evidence Q has been obtained). Applying Bayes theorem, we can write $p(E_i|Q) = p(Q|E_i)p(E_i)/p(Q)$. Since the query prior $p(Q)$ is merely a constant (the query does not change), we exclude it from further consideration [1]. The expression

$$p(E_i|Q) \propto p(Q|E_i)p(E_i) \tag{4.3}$$

[1]The denominator in any Bayesian formalism hardly enters into consideration. It is a normalising constant i.e. $p(Q) = \sum_{E_i} p(Q|E_i)p(E_i)$ and can be computed by requiring that $\sum_{E_i} p(E_i|Q) = 1$ (the posteriors over all of the variable's states to sum to unity) (Pearl, 1988; Korb and Nicholson, 2003).

forms the basis of the network shown in Figure 3.3 and establishes the underlying foundation of the ranking methodology. The process proceeds by unfolding the equation and inferring its parts via probabilistic inference, assuming the structure of our inference network: bottom-up belief propagation for the likelihood $p(Q|E_i)$ and top-down propagation for the priors $p(E_i)$. The two inferences form the essence of entity-diagnosis and entity-prediction that we wish to concur in the ranking. The inference process will consider a single instantiation state for each intermediate variable involved in the calculations, according to the conditions outlined in Chapter 3.

4.3.1 Top-down, predictive inference

Top-down propagation will not be fused in the inference at this point, and the network will remain clamped at each entity node on every evaluation. We achieve this by setting a uniform prior to the entities being evaluated. Hence $p(E_i)$ will be a constant for every entity node. In future extensions, we may introduce probabilities of the form $p(E_i|e)$ to simulate prior link analysis of entities, or propagation based on evidence from *backlink* information in the knowledge base, assuming crisp probabilistic values in the range $[0, 1]$. Another possibility is the modelling of a dataset dependency, such that to introduce the significance or popularity of the dataset as a factor in the inference (a prior piece of evidence). These possibilities are explored in the final chapter of the thesis.

4.3.2 Bottom-up, diagnostic inference

Diagnostic evidence reaching an entity member emanates from two sources of evidence: what we have defined in the previous chapter as the *local datatype context* and the *local object context* of each entity. Recall that we will be evaluating entities separately, with each consuming a distinct propagation process. The topological properties of the two contexts are very similar, except the variables involved are different. In a local

datatype context we have a variable number of term nodes linked to any number of local datatype property nodes, in turn linked to a single entity member. Similarly, in a local object context we have a variable number of entities (connected to their own local datatype contexts) connected to any number of local object property nodes, in turn linked to a single entity member. If we consider $L_1 = \{d_{i,1}, d_{i,2}, ..., d_{i,u}\}$ and $L_2 = \{o_{i,1}, o_{i,2}, ..., o_{i,v}\}$ to be the set of local datatype and object property nodes descending from entity E_i, respectively, then the diagnosis of E_i given a query specification may be encoded as:

$$p(Q|E_i) = p(L_1, L_2|E_i) \tag{4.4}$$

Assuming independence of the two sources, we are looking to evaluate the concurrence of the two estimates, $p(L_1|E_i)$ and $p(L_2|E_i)$, which can later be encoded into a weighted disjunction operator for more flexible retrieval.

4.3.2.1 Diagnosis of a single datatype property node

The diagnosis reaching a single local datatype property $d_{i,u}$ is delivered via a subset of term nodes $\{t_i, ..., t_n\} \in U$ descending from the respective property. Term nodes are diagnosed and instantiated with evidence in the query, and there is utmost a 1-to-1 mapping between query nodes in q_1 and the LS. Considering the independence assumptions of Section 4.2.1, the diagnosis of $d_{i,u}$ is defined as $\lambda(d_{i,u}) = \prod_j \lambda(t_j)\ p(t_j|d_{i,u})$, where the likelihood of a single term $\lambda(t_j)$ includes any evidence from the query layer q_1. We wish to approximate this estimate with a single canonical function that will give us flexibility in defining each individual conditional probability, in such a way that non-probabilistic values would also be applicable.

We can extrapolate the diagnosis of $d_{i,u}$ from evidence in the query as the cosine of the angle between two vectors (a vector of terms associated with q_1 and a vector of terms descending from the local datatype property). The specification is a valid and consistent

assumption because the cosine of two vectors is a number between 0 and 1, allowing propagation to continue further up the network. We define

$$\lambda(d_{i,u}) = \frac{\sum_{j=1}^{t} W_{t_j,d_{i,u},q_1} \times W_{t_j,d_{i,u}}}{\sqrt{\sum_{j=1}^{t} W^2_{t_j,d_{i,u},q_1}} \times \sqrt{\sum_{j=1}^{t} W^2_{t_j,d_{i,u}}}} \tag{4.5}$$

where $W_{t_j,d_{i,u}}$ was defined earlier in Section 4.2.1 and can feature the effect of an indexing weight, with respect to the given property. $W_{t_j,d_{i,u},q_1}$ is a function of the likelihood of a query term node (or the diagnosis of a term accorded by evidence in the query) and can feature any form of frequency measure, such as the inverse collection frequency (IDF)[2]. The denominator in Equation 4.5 depicts the product of the two vectors' Euclidean length, and the summations are over all the terms associated with each vector. Intuitively, we have made $p(q_1|d_{i,u})$ equivalent to $cos(\vec{q_1}, \vec{d_{i,u}})$. The specification is consistent with the topology of the network, since a cosine measure otherwise computes the degree to which the concept $d_{i,u}$ is covered by the query layer q_1 and retains a probabilistic character that allows us to engage in further propagation.

4.3.2.2 Diagnosis of an entity from its local datatype context

An entity member is connected to a variable number of local datatype property nodes, each communicating diagnostic support to the member variable. In other words, an entity member will receive a series of diagnostic messages emanating from property nodes that have at least one of their descendants (term nodes) activated by the query. We enclose these messages into a disjunction operator to ensure that evidence from multiple properties will increase and not diminish the final diagnosis. We define

$$p(L_1|E_i) = 1 - \prod_{u}(1 - \lambda(d_{i,u})p(d_{i,u}|E_i)) \tag{4.6}$$

[2]The precise weighting schemes implemented and utilised during evaluation of the model are covered in the following chapter.

to be the diagnosis of an entity, given its local datatype context and the evidence of a query. The conditional probability $p(d_{i,u}|E_i)$ was defined earlier in Section 4.2.2 and acts as a filter on each diagnostic message. The form of disjunction in the equation enables diagnostic messages from any one or more local datatype property nodes to be a sufficient condition for activating and retrieving the entity member. During the actual computation, we need only consider properties that have at least one descendant activated by the query.

4.3.2.3 Diagnosis of a single object property node

The diagnosis reaching a single local object property $o_{i,v}$ is delivered via a subset of entity nodes $\{E_i, ..., E_n\}$ descending from the respective property that have been activated by the query. This can potentially result in a loopy inference process, since an entity's local datatype diagnosis $p(L_1|E_i)$ can end up being double counted if delivered back from connected entities via its local object context. We constrain propagation to a single "hop" in the network to refrain this from happening. We can potentially explore multiple hops in a more controlled environment, but we will not be investigating the possibility at this point.

A single local object property node can be linked to multiple entities in the network e.g. the case of a person who knows many other persons. At the same time a single entity node can be linked to several local object property nodes of a given entity e.g. the case of a person who knows and works with another person. We will enclose diagnostic messages resulting from this topology into a double disjunction operator, in effect allowing diagnosis from any number of entities and any number of object properties to be sufficient for enhancing the probability of a given entity. This is illustrated in Figure 4.1. We define

$$\lambda(o_{i,v}) = 1 - \prod_j (1 - p(L_1|E_j)p(E_j|o_{i,v})) \tag{4.7}$$

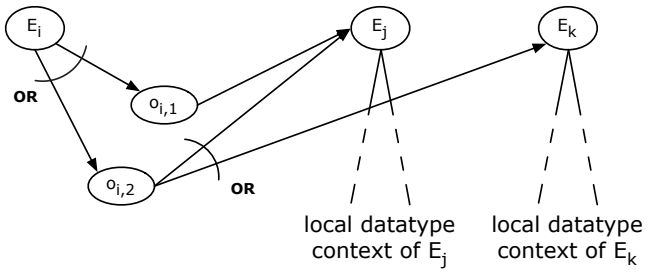


Figure 4.1: A 2-level disjunction operator (Noisy-OR gates) enclosed in the diagnosis of a member variable via its local object context.

to be the diagnosis of property $o_{i,v}$ delivered from a variable number of entities that have their local datatype contexts activated by the query (the rightmost operator in Figure 4.1).

4.3.2.4 Diagnosis of an entity from its local object context

An entity member is connected to a variable number of local object property nodes, each communicating diagnostic support to the member variable (as in Figure 4.1). We use the same strategy, as done previously for local datatype properties, to capture these diagnostic messages into a disjunction operator and ensure that evidence from multiple properties will increase and not diminish the final diagnosis. We define

$$p(L_2|E_i) = 1 - \prod_{v}(1 - \lambda(o_{i,v})p(o_{i,v}|E_i)) \tag{4.8}$$

to be the diagnosis of an entity, given its local object context. The conditional probability $p(o_{i,u}|E_i)$ was defined earlier in Section 4.2.2 and acts as a filter on each diagnostic message. Equation 4.8 encloses a 2-level disjunction operator to accommodate the arbitrary connection of entities and the local object properties of the member being evaluated. This is further illustrated in Figure 4.1.

4.3.2.5 Putting it all together

We complete the specification by enclosing the two sources of evidence (L_1 and L_2) into a disjunction operator, such that to allow flexible matching given any one of the two specifications. When evidence from both sources is available, the final diagnosis will increase but not diminish. The unified diagnostic inference formula[3]:

$$p(L_1, L_2|E_i) = 1 - \left(\prod_u 1 - \lambda(d_{i,u})p(d_{i,u}|E_i)\right)\left(\prod_v 1 - \lambda(o_{i,v})p(o_{i,v}|E_i)\right) \quad (4.9)$$

is applicable to every entity being activated by the query (according to the conditions defined in Chapter 3). Ranking is constrained to only involve propagation emanating from the query via the local datatype context of E_i and the local datatype contexts of entities linked directly from the member's local object context.

[3]The formula is the result of a simple factorisation e.g. $1-(1-(1-(1-x)))(1-(1-(1-y))) = 1-(1-x)(1-y)$.

Chapter 5

Worked Example

This chapter provides a summary and visual walkthrough of the various aspects of the model presented in the previous two chapters (translation from a directed labelled graph, observing a query, assignment of probabilities, and ranking via probabilistic inference). It is important to read Chapters 3 and 4 before embarking on this one, as we will not be re-introducing concepts covered previously. Several pointers, however, are provided throughout the walkthrough. Some implementation details are also provided, along with pointers to the following chapter. The following chapter presents an implementation and evaluation of the model on a realistic dataset and a manually constructed set of queries. Aspects such as precise frequency measures, parameter tuning, and text processing are covered in the following chapter, but also mentioned hereafter.

5.1 Translating the Data Graph

Figure 5.1 presents the sample triple graph that we will use to model as a Bayesian Network for entity search. We are now looking to translate this data into a Resource Network (Section 3.4). We identify the following variables to translate as nodes in the Resource Network:

- 3 Entity Members (Section 3.4.1): `this:record1`, `this:record2`, `this:tag1`

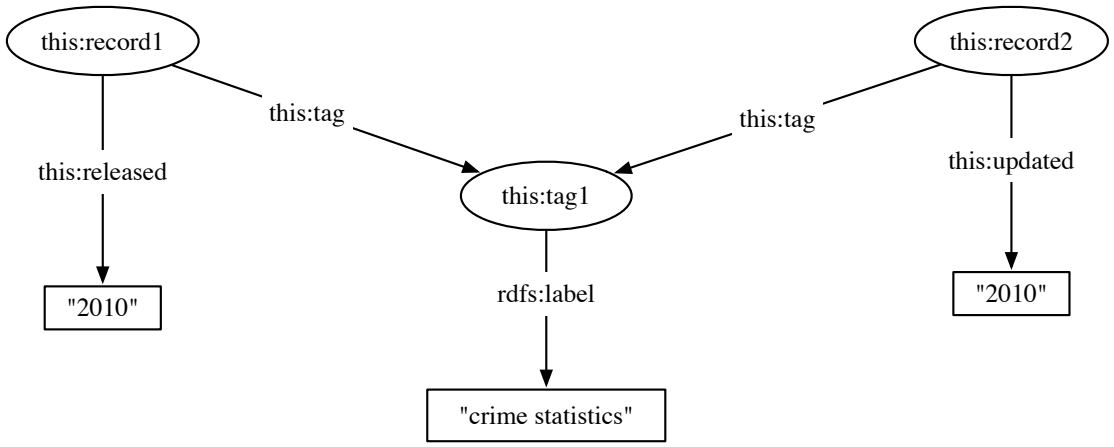


Figure 5.1: Example data graph.

- 4 Global Property Nodes (Section 3.4.2.1): `this:tag`, `this:released`, `rdfs:label`, `this:updated`
- 9 terms for the Literal Space (Section 3.4.3): *2010, released, label, updated, tag, crime, statistics, record1, record2*

Note that we have included 6 candidate terms for the Literal Space that are only part of the URIs of resources. The only literals in the knowledge base are "2010" and "crime statistics". The rest are induced from the URIs. As we have indicated in Chapter 3, when a resource that is candidate for either a global property node or entity member does not have an associated label (`rdfs:label`), then we extract terms from the URI of the resource and use those as the resource's `rdfs:label`.

From the 4 global property nodes, we define 5 *local property nodes* for the Resource Network, as explained in Section 3.4.2.2 (local property nodes reflect the usage/appearance of properties in what we defined as the *local datatype* and *local object contexts* of entity members - defined in Section 3.4.1). Table 5.1 presents the correspondence/mapping between variables in the data graph and nodes in the Resource Network. From this arrangement, and according to the rules and conditions explained throughout Chapter 3 (in the respective sections of each type of variable), we can now develop the Resource

Data	Translated Variable	Data	Translated Variable
`this:record1`	E_1	*record1*	t_2
`this:tag1`	E_2	*released*	t_3
`this:record2`	E_3	*tag*	t_4
`this:tag`	O_1	*crime*	t_5
`this:released`	D_1	*statistics*	t_6
`rdfs:label`	D_2	*label*	t_7
`this:updated`	D_3	*updated*	t_8
2010	t_1	*record2*	t_9

Table 5.1: Translation of data to variables/nodes for the Bayesian Network.

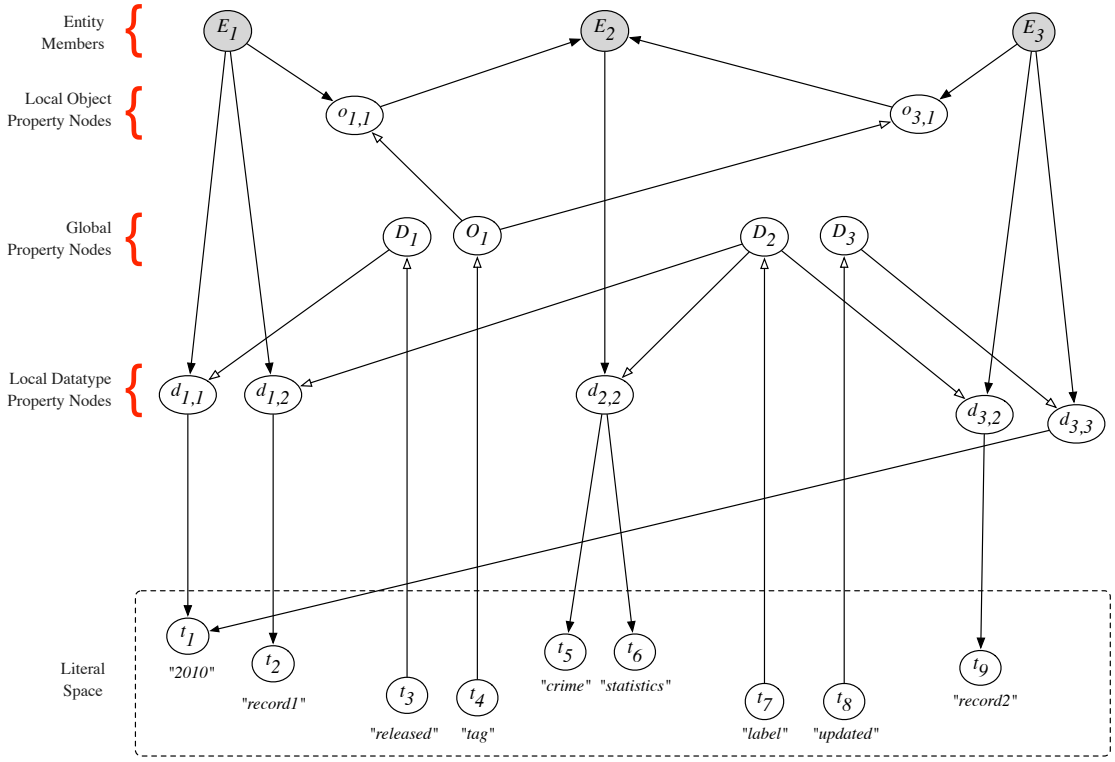


Figure 5.2: Translated data to the Bayesian Network model.

Network. The outcome is illustrated in Figure 5.2. Local datatype and object property nodes are defined and named accordingly to reflect their parents e.g. a local datatype property with parents E_1 and D_2 is named $d_{1,2}$ accordingly - Section 3.4.2.2).

The diagram may look complicated in its current state, but recall that not all nodes will participate in propagation. When a query is observed and the instantiation of *local* property nodes occurs, all *global* property nodes and their associated links will be removed from the diagram. Propagation only takes place within the local contexts of entity members (through local object and datatype property nodes). In later figures we will use a simpler diagram to outline the inference process.

Most of the mathematical expressions for conditional dependencies in the model are carried implicitly. Since we are looking at a Bayesian Network, anyone should be able to identify that the conditional probability $p(d_{1,1}|E_1)$ exists between the nodes $d_{1,1}$ and E_1. See our review of Bayesian Network theory in Section 3.2.

However, as we have outlined and explained in detail in Chapter 4, we develop several assumptions in the model that have a direct influence on how we treat/evaluate and quantify some of the conditional dependencies. Specifically, complexities can arise from the dependencies of term nodes to the local datatype property nodes, and entity members to the local object property nodes of other members. We revisit these two cases before we proceed.

5.1.1 Resolving complex dependencies of term nodes

As explained in Section 4.2.1, potentially complex dependencies between term nodes and local datatype property nodes, such as $p(t_x|d_{i,j}, d_{i,k})$ (depicting a term node linked to two local datatype property nodes of the *same* entity member E_i - conceptually - not present in the current diagram) are approximated by a higher order prototypical function that produces a different value for each individual dependency. This assumes independence of property nodes, which, according to the rules of conditional independence in Bayesian Networks (see Bishop et al., 2006, $3rd$ case on page 375), means that term nodes are never actually instantiated during inferencing. This is not necessarily true, since we will encapsulate and generalise the instantiation of term nodes in a scalar product formula

at the property node (Section 4.3.2.1). This is in accordance to our two assumptions defined in Section 4.2.1 regarding term nodes. Furthermore, by using the prototypical function in Section 4.2.1 we avoid having to enumerate in a contingency table all possible probabilities of term nodes. This would not be possible, since we do not know in advance the organisation of the data, which can be arbitrary.

Also note that the complexity that may result from the association of term nodes to local datatype property nodes associated with different entity members (such as $p(t_1|d_{1,1}, d_{3,3})$ in Figure 5.2) will not be present in the inference process, since entity members are treated in *isolation* against the query (as customary with any IR model). This is clarified in both Chapters 3 and 4.

5.1.2 Resolving complex dependencies of entity members

Similarly to the case of term nodes mentioned above, a potential complexity may arise when an entity member is linked to two or more local object property nodes of another entity member e.g. a probability of the form $p(E_j|o_{k,1}, o_{k,2})$ in Figure 3.3 of Chapter 3. We do not make a case to enumerate all possible values for this complex dependency, since it would require exhaustive enumeration of parent values, something not possible with an arbitrary translation of data. As done with term nodes, we project these properties onto a prototypical function for each individual dependency link. We use a Bayesian filter ($p(E_i|o_{j,k}) = p(E_i \wedge O_k)/p(O_k)$ defined in Section 4.2.3 - Equation 4.2) to quantify each link separately. This has been justified and plays an important role in estimating the importance of object properties as appearing in the definition of entities.

Similarly to term nodes, the complexity that may result from the association of entity members to the local object property nodes of two or more different entity members (such as $p(E_2|o_{1,1}, o_{3,1})$ in Figure 5.2) will not be present in the inference process. Entity members will be treated in isolation, causing the network to be clamped around their local object and datatype contexts. Therefore, although the dependency $p(E_2|o_{1,1}, o_{3,1})$

Dependency	Quantified As...	Value
$p(E_2 = true \| o_{1,1} = true)$	$p(E_2 \wedge O_1)/p(O_1)$	$2/2 = 1$
$p(E_2 = true \| o_{1,1} = false)$	$p(E_2 \wedge O_1)/p(O_1)$	$2/2 = 1$
$p(E_2 = true \| o_{3,1} = true)$	$p(E_2 \wedge O_1)/p(O_1)$	$2/2 = 1$
$p(E_2 = true \| o_{3,1} = false)$	$p(E_2 \wedge O_1)/p(O_1)$	$2/2 = 1$

Table 5.2: The conditional dependencies of entity members, generalised and quantified with the Bayesian filter from Equation 4.2.

exists, only $p(E_2|o_{1,1})$ will be present when E_1 is being evaluated, and only $p(E_2|o_{3,1})$ when evaluating E_3.

At this point we have enough information to evaluate the Bayesian filter (Section 4.2.3 - Equation 4.2) for the member dependencies. This value is precomputed and we can pull it out when we propagate evidence across entities. In our example, the Bayesian filter gives a value of 1 for both $p(E_2|o_{1,1})$ and $p(E_2|o_{3,1})$. The filter is computed once per dependency and does not vary on the states of the local property nodes. The filter is only used to propagate evidence from entity members instantiated to $true$ later by the query. Members set to $false$ do not take part in propagation. The values of the Bayesian filter are provided for clarity in Table 5.2.

5.2 Observing a Query and Instantiating the Network

Before we proceed with quantifying and putting the actual values of the dependencies and the nodes on the network, it is important to briefly re-instate how we treat and reason with them and the instantiation conditions of the variables. As clearly stated by now, we do not aim to evaluate or differentiate all possible instantiation states of the variables in the diagram, but rather only manipulate sets of propositions and their states. We use

prototypical functions that are practical and easier to implement (such as the Bayesian filter for the dependencies $p(E_i|o_{j,k})$ - Equation 4.2), and we do not consider the false states of entity members (Section 3.4.1), since an entity member set to $false$ means that it contains no diagnostic path to the query.

We can now proceed with observing a query. We will use query evidence to instantiate the Resource Network, quantify the conditional dependencies that will take part in propagation (not all of them will e.g. probabilities associated with term nodes will be generalised in a cosine rule), and infer its impact on the entities that are candidate for retrieval.

5.2.1 Query nodes and layers

A query for "*crime statistics released in 2010*" results in a Query Network being attached to the Resource Network for propagation, as shown in Figure 5.3. See Section 3.5 - The Query Network - for precise details on what the two layers in the Query Network mean. The basic idea is that all terms in the query that have an equivalent (contextually identical) term in the Literal Space (terms that nodes in the Literal Space represent) will be included in a Query Network. Inside layer ***q2*** we include all terms that have an equivalent node in the Literal Space that *ascends* from a global datatype or object property node. Inside layer ***q1*** we include all terms that have an equivalent node in the Literal Space that *descends* from a local datatype property node.

In this example, query nodes and nodes in the Literal Space have a one-to-one correspondence without requiring any text processing. In the following chapter, when we deal with real data, both the terms represented by nodes in the Literal Space and the terms represented by nodes in the Query Network will go through the same text processing to increase the chances of finding these mappings.

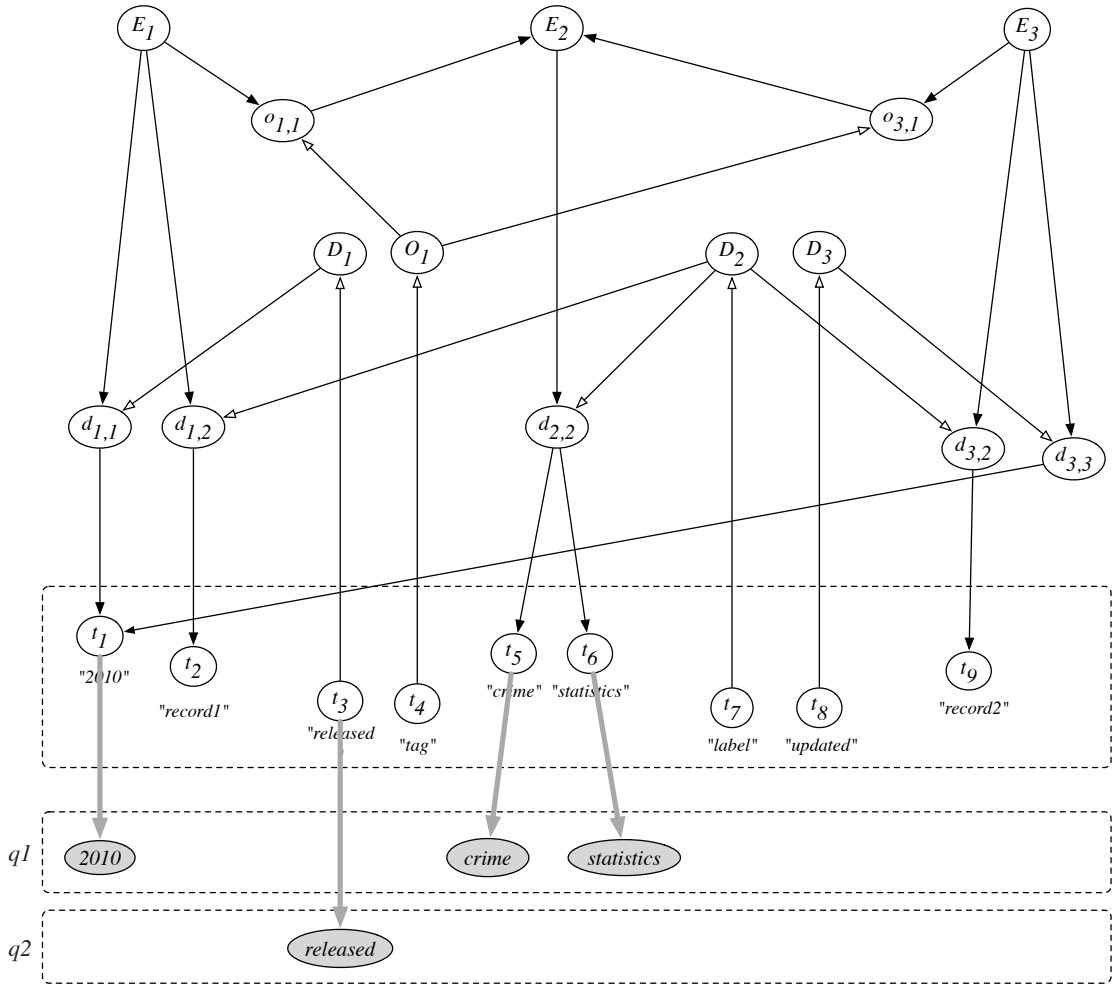


Figure 5.3: Example Query Network attached to the Resource Network for the query "crime statistics released in 2010".

5.2.2 Inferring the impact of evidence from the query

5.2.2.1 Evaluating *q2*

The first layer that we evaluate is $q2$. This layer contains the *property evidence* that we discussed thoroughly in Chapter 3 - Section 3.5.2. It is meant to set the instantiation of global, and hence local, datatype and object property nodes. There is a parameter associated with the instantiation process (param. γ in Equation 3.6), which for this experiment we set to 0. As such, any mapping between $q2$ and a global property node

(i.e. any one or more terms matching) will set the state of the property to $true$. In this example, D_1 is set to $true$. Consequently, all local properties descending from it will also be set to $true$ (Section 3.4.2.2 - Equation 3.5). The effect of this is asserting on the network that $d_{1,1} = true$ (Figure 5.4). All other global and local properties will be set to $false$.

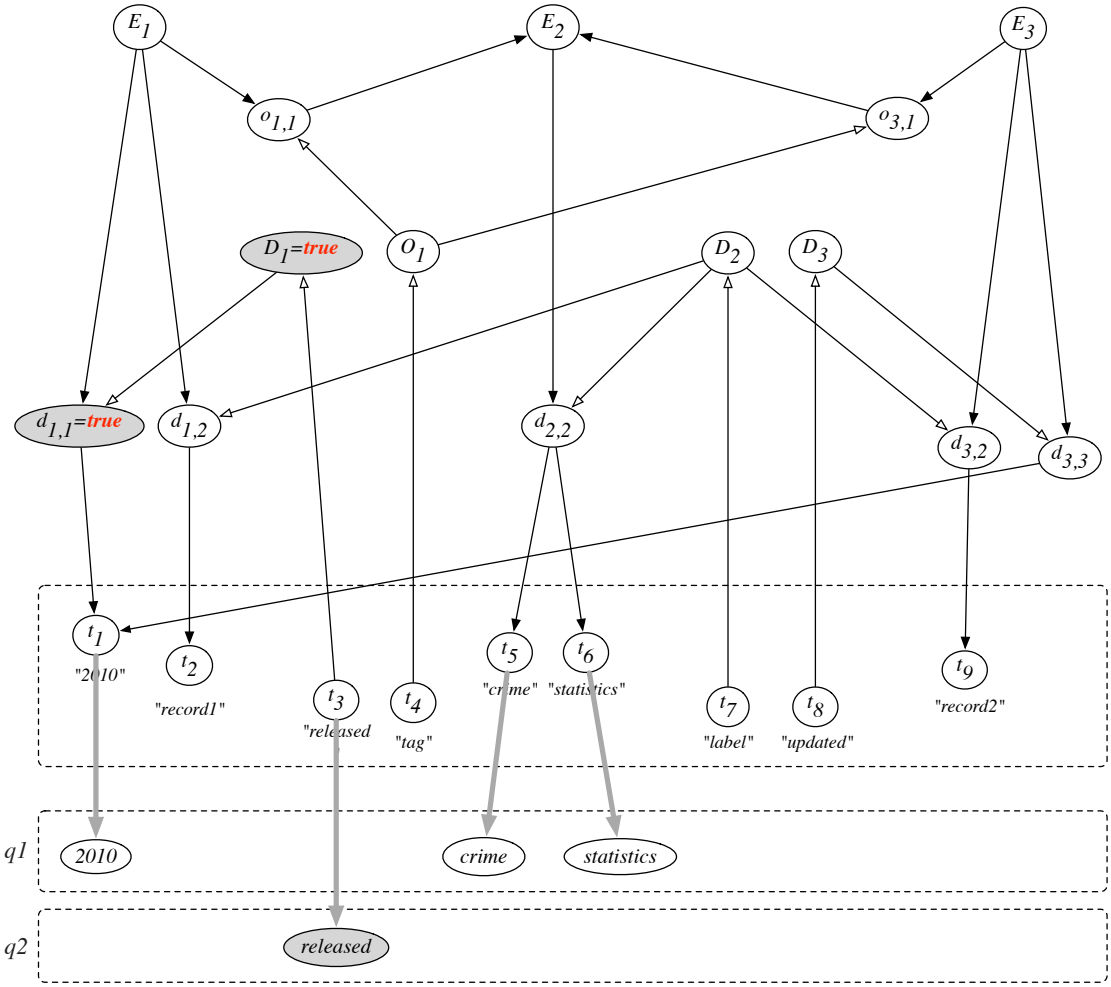


Figure 5.4: Evaluating the query layer *q2* and asserting that a global and local datatype property nodes (D_1 and $d_{1,1}$ resp.) are set to *true*.

Once we know the instantiation of local property nodes, we can discard $q2$ from the process. We can also remove the global property nodes from the diagram from this point onward. Global property nodes are only needed to compute the Bayesian filter

(Section 4.2.3 - Equation 4.2) we mentioned above, and to act as an efficient mapping to the Literal Space for instantiating local properties. Figure 5.5 shows what is left of the network after $q2$ has been observed and removed, the local properties instantiated to their $true/false$ states, and the global property nodes removed (along with term nodes that were only linked to them). Take the time to compare with Figure 5.3 above.

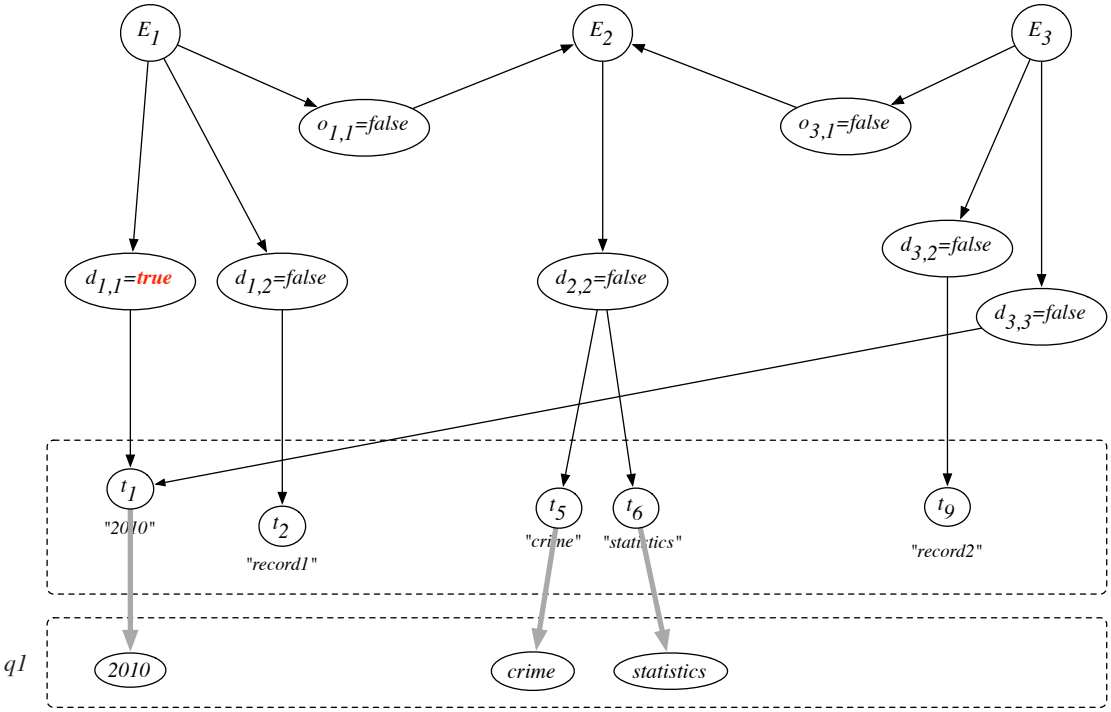


Figure 5.5: What is left of the network after *q2* has been evaluated. Local property nodes appear as instantiated to either *true* or *false* and global property nodes have been removed.

The conditional probabilities of local property nodes on entity members are the main methods for external parameterisation of the model during query time. These have been discussed in Sections 3.4.2.2 and 4.2.2 of the thesis. In the following chapter (Section 6.4.2.2), we discuss a maximum likelihood estimation (MLE) method to tune these parameters over a range of queries to achieve maximum performance on average. When tuning the parameters a priori, we achieve a set of values that will work satisfactorily over a given collection. Fine-tuning them per query though can embody different effects,

Dependency	Value
$p(d_{i,j} = true \| E_i = true)$	0.6
$p(d_{i,j} = false \| E_i = true)$	0.4
$p(o_{i,j} = true \| E_i = true)$	0.6
$p(o_{i,j} = false \| E_i = true)$	0.4

Table 5.3: The dependencies of local datatype and object property nodes on entity members, assigned/quantified to a set of manually defined weights.

depending on the query. We have discussed these in the aforementioned sections. For this example, we set the probabilities to the values outlined in Table 5.3. This would be the same to setting q in Equation 4.1 to 0.6 for both datatype and object properties.

Note that in Table 5.3 we are only interested in outlining the probabilities corresponding to the *true* states of entity members ($E_i = true$). The values would be the same for both states, except members set to *false* will not engage in propagation. There is therefore no need to overflow the table with repeated values that will not be used.

5.2.2.2 Evaluating *q1*

When observing $q1$, the first thing that occurs is the instantiation of entity members. An entity member is set to $true$ when it has a diagnostic path open to the query layer (Section 3.4.1). A diagnostic path can run through either its *local datatype content* or the local datatype contexts of other members connected to it via its *local object context* (Section 3.4.1). All three entity members in our diagram are instantiated to $true$. This is illustrated in Figure 5.6. We have also put the probabilities from the contingency tables we know so far on the diagram for clarity.

At this point we are looking to evaluate the impact of the evidence in $q1$ on the local datatype contexts of entity members. This resolves to measuring the diagnostic

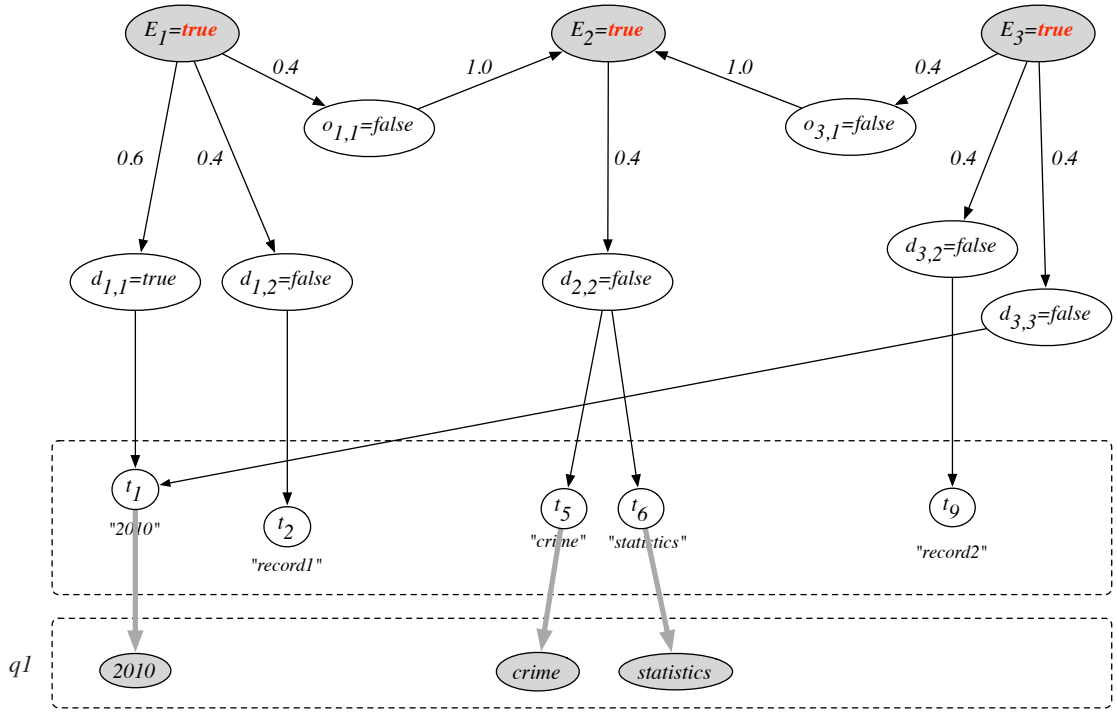


Figure 5.6: Entity members instantiated to *true*. Known conditional probability values placed over their corresponding links.

(bottom-up) support that the query layer can provide to each local datatype property node ($d_{i,j}$) in the diagram. As stated previously, we use a cosine rule to approximate this diagnosis as it passes through the Literal Space in the diagram. Figure 5.7 marks an example of this type of diagnostic support using the notation we defined in Section 4.3.2.1.

The cosine rule is defined in Section 4.3.2.1, Equation 4.5. As stated in the respective section, we are effectively making $p(q_1|d_{i,j})$ equivalent to $cos(\vec{q_1}, \vec{d_{i,j}})$. This is a consistent assumption, inline with our two assumptions about the co-dependencies of term nodes, defined in Section 4.2.1. The cosine rule is the backbone of the Vector Space Model in traditional IR (Manning et al., 2008; Baeza-Yates and Ribeiro-Neto, 1999), among the most commonly used techniques, and allows us to generalise the instantiation of term nodes.

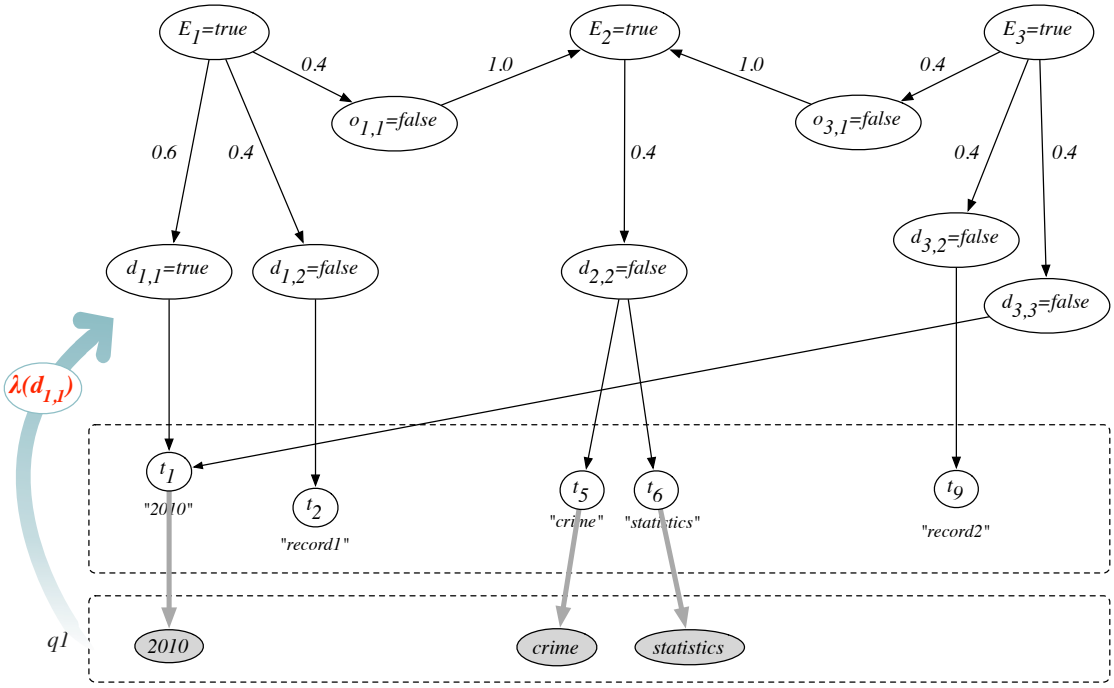


Figure 5.7: A visual depiction of the diagnosis of a single datatype property node - Section 4.3.2.1.

We need two variables to compute the cosine rule: $W_{t_j,d_{i,u}}$ and $W_{t_j,d_{i,u},q_1}$. These have been defined in Sections 4.2.1 and 4.3.2.1 respectively, although precise frequency measures are given in the following chapter when we implement the model over a realistic collection (Section 6.4.2.1). In Section 6.4.2.1 we quantify these estimates according to the traditional Term Frequency (TF) and Inverse Collection Frequency (IDF) weights, with the added expressivity of property-level frequency information. We use the same methods outlined in Table 6.4 to quantify the estimates in our example here. All the corresponding computations are given in Table 5.4.

Based on the values from Table 5.4, we now have the diagnostic support accorded to each local datatype property node from the query. We can place these in the diagram and remove all the unnecessary links and nodes. Figure 5.8 puts the diagnostic messages in the diagram and discards the query layer $q1$ and the Literal Space. We now have all the pieces in place to proceed with the remaining inference.

Variable	Quantified As...	Value
$W_{t_1,d_{1,1}}$	$1 + log(tf_{t_1,d_{1,1}})$	$1 + log(1) = 1.0$
$W_{t_1,d_{3,3}}$	$1 + log(tf_{t_1,d_{3,3}})$	$1 + log(1) = 1.0$
$W_{t_2,d_{1,2}}$	$1 + log(tf_{t_2,d_{1,2}})$	$1 + log(1) = 1.0$
$W_{t_5,d_{2,2}}$	$1 + log(tf_{t_5,d_{2,2}})$	$1 + log(1) = 1.0$
$W_{t_6,d_{2,2}}$	$1 + log(tf_{t_6,d_{2,2}})$	$1 + log(1) = 1.0$
$W_{t_9,d_{3,2}}$	$1 + log(tf_{t_9,d_{3,2}})$	$1 + log(1) = 1.0$
$W_{t_1,d_{1,1},q_1}$	$1 + log\frac{N_{E,D_1}}{df_{t_1,E,D_1}}$	$1 + log(1/1) = 1.0$
$W_{t_1,d_{3,3},q_1}$	$1 + log\frac{N_{E,D_3}}{df_{t_1,E,D_3}}$	$1 + log(1/1) = 1.0$
$W_{t_5,d_{2,2},q_1}$	$1 + log\frac{N_{E,D_2}}{df_{t_5,E,D_2}}$	$1 + log(3/1) = 1.477$
$W_{t_6,d_{2,2},q_1}$	$1 + log\frac{N_{E,D_2}}{df_{t_6,E,D_2}}$	$1 + log(3/1) = 1.477$
$\lambda(d_{1,1})$	$cos(\vec{q_1}, \vec{d_{1,1}})$	$\frac{1.0 \times 1.0}{\sqrt{1.0^2} \times \sqrt{1.0^2}} = 1.0$
$\lambda(d_{1,2})$	$cos(\vec{q_1}, \vec{d_{1,2}})$	0.0
$\lambda(d_{2,2})$	$cos(\vec{q_1}, \vec{d_{2,2}})$	$\frac{(1.0 \times 1.477) + (1.0 \times 1.477)}{\sqrt{1.477^2 + 1.477^2} \times \sqrt{1.0^2 + 1.0^2}} = 1.0$
$\lambda(d_{3,2})$	$cos(\vec{q_1}, \vec{d_{3,2}})$	0.0
$\lambda(d_{3,3})$	$cos(\vec{q_1}, \vec{d_{3,3}})$	$\frac{1.0 \times 1.0}{\sqrt{1.0^2} \times \sqrt{1.0^2}} = 1.0$

Table 5.4: Variables and their quantities for computing the diagnosis of each local datatype property node in the example.

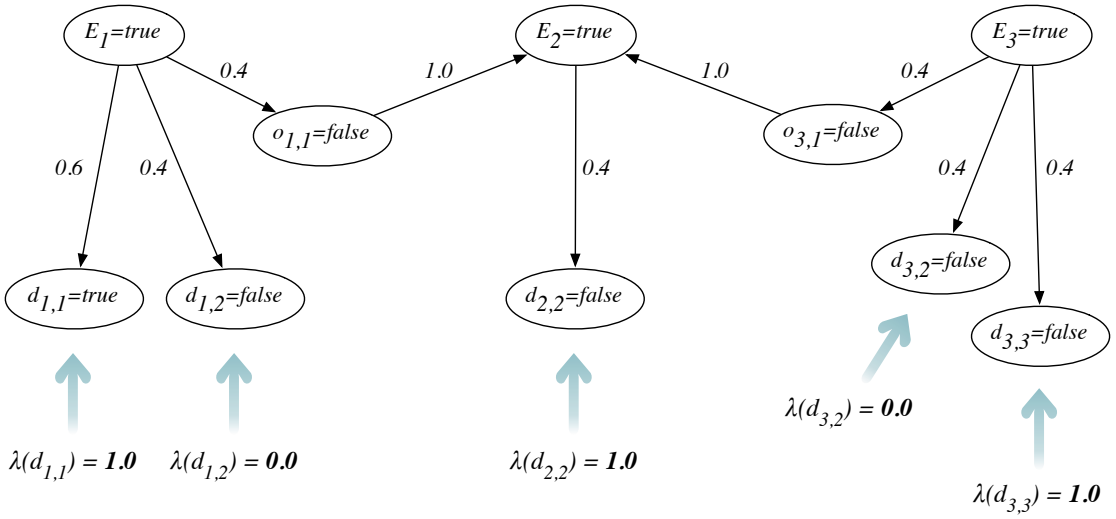


Figure 5.8: A visual depiction of the diagnostic support accorded to each local datatype property node from the query. The query layer *q1* and the Literal Space have been removed from the diagram as they have been evaluated.

5.3 Completing the Inference and Retrieval

At this point, all the links in the network have been quantified to specific values, and the diagnosis of local datatype property nodes has been computed. We can now proceed with propagating these diagnostic messages to the entity members to compute their probability for retrieval. As explained in Chapter 4, the diagnostic support that local datatype and object property nodes provide to entity members are combined via disjunction at the entity members. These disjunction operators form an essential part of our ranking strategy and are evident in all of the diagnostic formulas in Section 4.3.2. With the disjunction of messages reaching entity members, we allow diagnosis from any single datatype or object property node to be a sufficient condition for retrieving the respective entity. The disjunction operators are illustrated as logical OR gates in Figure 5.9 (even when there is only a single link - hence the logical condition will have no effect).

We now proceed with computing the diagnosis of each individual entity from its local datatype context (Section 4.3.2.2). We treat each entity in isolation, hence entity

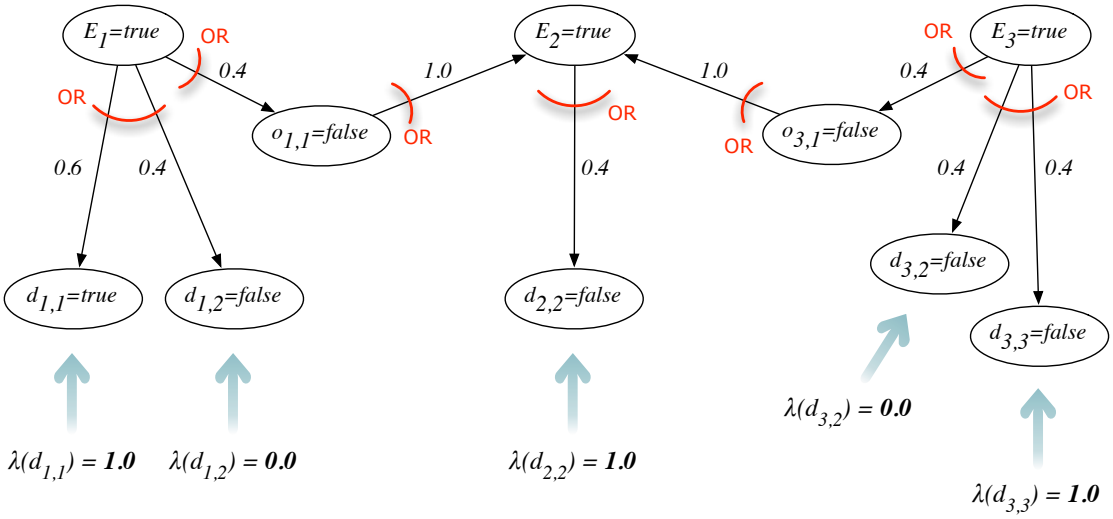


Figure 5.9: Disjunction operators marked as logical OR gates in the diagram.

members diagnosed with positive evidence from the query (λ messages in Figure 5.9) are evaluated iteratively one by one. The network at this point becomes clamped around the local datatype contexts of each entity. This is illustrated in Figure 5.10. We are looking to complete the evaluation of $p(L_1|E_i)$ (Section 4.3.2) for each entity member. The diagnostic formula for accumulating the messages reaching an entity member from its local datatype context is given in Equation 4.6. Table 5.5 presents the results of Equation 4.6 evaluated for each entity member in the diagram.

Having computed the diagnostic support that each entity member receives from the local datatype property nodes, we can proceed with propagating these messages to connected entities in the network. At this point, the diagram has been simplified to that of Figure 5.11. The local datatype contexts have been evaluated and replaced with the diagnostic messages $p(L_1|E_i)$. The next step is to evaluate the local object contexts of entity members (the second part of Equation 4.4 - $p(L_2|E_i)$). The inference procedures associated with this form of diagnosis have been presented across two sections in the thesis: Section 4.3.2.3 - titled *Diagnosis of a single object property node*, and Section 4.3.2.4 - titled *Diagnosis of an entity from its local object context*.

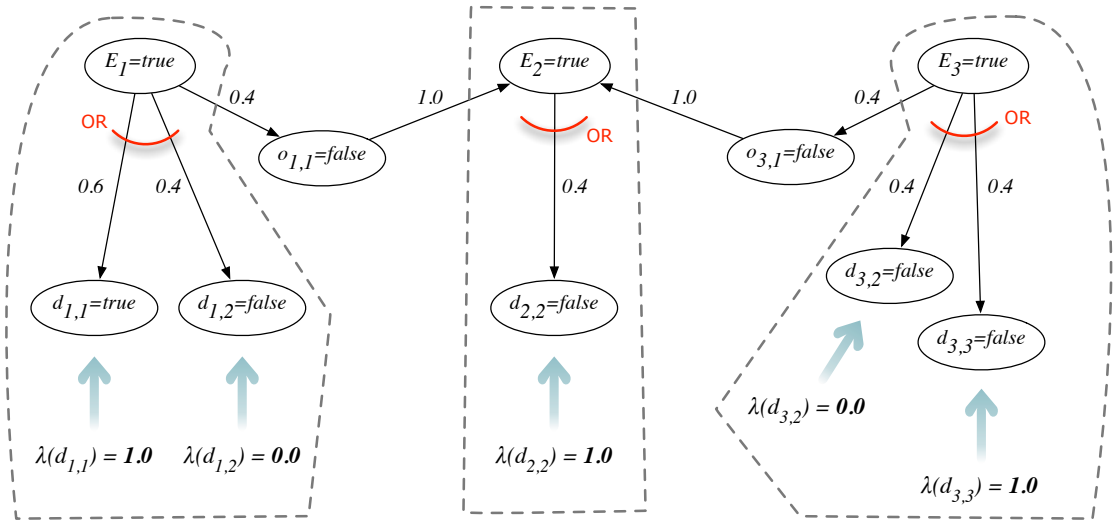


Figure 5.10: The local datatype contexts of the three entity members.

Variable	Quantified As...	Value
$p(L_1\|E_1)$	$1 - \prod_u (1 - \lambda(d_{1,u})p(d_{1,u}\|E_1))$	$1 - [(1 - (1 \times 0.6)) \times (1 - 0)] = 0.6$
$p(L_1\|E_2)$	$1 - \prod_u (1 - \lambda(d_{2,u})p(d_{2,u}\|E_2))$	$1 - (1 - (1 \times 0.4)) = 0.4$
$p(L_1\|E_3)$	$1 - \prod_u (1 - \lambda(d_{3,u})p(d_{3,u}\|E_3))$	$1 - [(1 - 0) \times (1 - (1 \times 0.4))] = 0.4$

Table 5.5: The computed diagnoses of entity members from their local datatype contexts.

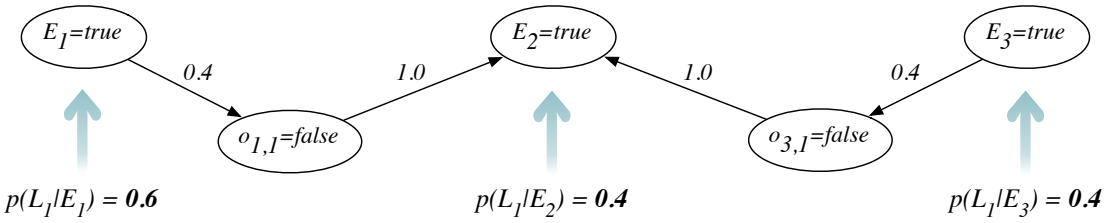


Figure 5.11: The local datatype contexts of the three entity members removed and replaced by the computed probabilities from Table 5.5.

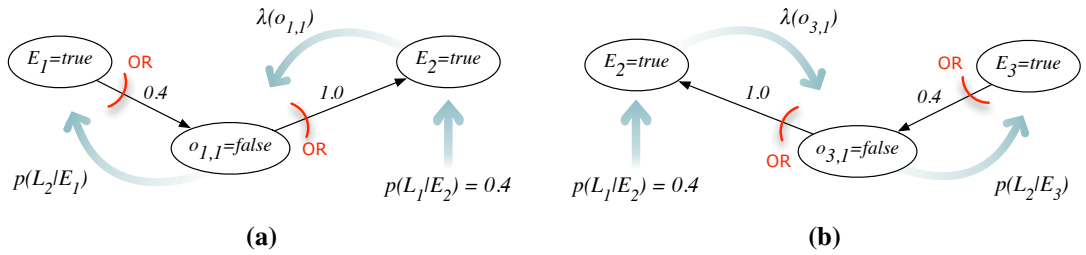


Figure 5.12: A visual illustration of how diagnostic messages are transmitted from entity member E_2 to each of E_1 and E_3 via their local object contexts.

Variable	Quantified As...	Value
$\lambda(o_{1,1})$	$1 - \prod_j (1 - p(L_1\|E_j)p(E_j\|o_{1,1}))$	$1 - (1 - (0.4 \times 1)) = 0.4$
$\lambda(o_{3,1})$	$1 - \prod_j (1 - p(L_1\|E_j)p(E_j\|o_{3,1}))$	$1 - (1 - (0.4 \times 1)) = 0.4$
$p(L_2\|E_1)$	$1 - \prod_v (1 - \lambda(o_{1,v})p(o_{1,v}\|E_1))$	$1 - (1 - (0.4 \times 0.4)) = 0.16$
$p(L_2\|E_3)$	$1 - \prod_v (1 - \lambda(o_{3,v})p(o_{3,v}\|E_3))$	$1 - (1 - (0.4 \times 0.4)) = 0.16$

Table 5.6: Calculations associated with computing the diagnoses of entity members from their local object contexts.

As done previously, each entity member is treated separately for evaluation. One by one, entity members request from connected *activated* entity members (those connected to their local object contexts and instantiated to $true$ by the query) to send them the diagnosis they have accumulated in their local datatype contexts. In our example, this involves two entity member evaluations: one for E_1 and one for E_3, since E_2 has no local object context defined. Each of E_1 and E_3 will request from E_2 to transmit its diagnosis. The decoupling of the network and the process are illustrated in Figure 5.12, including the diagnostic formulas we are going to evaluate next.

Table 5.6 presents the results from evaluating Equations 4.7 and 4.8 from Chapter 4, as illustrated in Figure 5.12. Given these results, we now have all the necessary ingredients to compute the total probability of the entity members and complete our ranking strategy (Section 4.3). We proceed with evaluation of Equation 4.9, which depicts the disjunction of the two messages accumulated at the entity members: the diagnosis from their local datatype contexts ($p(L_1|E_i)$) and the diagnosis from their local object contexts ($p(L_2|E_i)$). The results form combining these messages are given in Table 5.7.

At this point, we have effectively ranked the three entities, leading to the order given in Table 5.7. The most relevant entity for the query is E_1 with probability 0.664 (66.4%), followed by E_3 with probability 0.496 (49.6%), and E_2 with probability 0.4 (40%).

Variable	**Value**
$p(L_1, L_2\|E_1)$	$1 - [(1 - 0.6) \times (1 - 0.16)] = 0.664$
$p(L_1, L_2\|E_3)$	$1 - [(1 - 0.4) \times (1 - 0.16)] = 0.496$
$p(L_1, L_2\|E_2)$	$1 - [(1 - 0.4) \times (1 - 0)] = 0.4$

Table 5.7: The results from combining the diagnostic messages accumulated at the entity members. These equate to the final ranking of the three entities.

Chapter 6

Model Instantiation and Effectiveness Evaluation

6.1 Introduction

In this previous three chapters, we have presented a network inference model for entity-oriented searches in semantic knowledge bases. Our focus was on the conceptual definition of the model down to precise orchestration of inference processes and extraction of probability distributions. In this chapter, we proceed with evaluation of the model to offer an overview of what it delivers when applied over a realistic data collection and to indicate its demonstrated performance. The focus of the evaluation is on the effectiveness of the model to reason and respond appropriately to the key retrieval features and aspects presented in the previous chapters.

In Section 6.2, we formulate a set of hypotheses to be approved or rejected via a series of evaluation experiments. The evaluation methodology is presented in Section 6.3, including details on the selected data collection, queries and assessments used for Precision/Recall analyses, and the precise evaluation metrics selected for the experiments. Section 6.4 presents implementation details, such as configuration parameters and other processing necessary to instantiate the model to an operational state for evaluation. Section 6.5 presents results from the evaluation experiments and hypothesis testing

using aggregated and per-query scores. Section 6.6 concludes the chapter with a general discussion on the choices made for the evaluation.

6.2 Evaluation Hypotheses

The scope of the evaluation is to assess the model's core set of features, in essence restricting emphasis to the fundamental aspects of the model and not the potential improvements that may result from viable extensions to the model (e.g. boosting document priors with link analysis). We do not focus on evaluating the effects of core conventional IR techniques either, such as use of stop-lists, TF-IDF weights, and stemming. These have been subject to many empirical tests in the literature and proven indispensable for any retrieval system. Additionally, Information Retrieval sequences can embody several distinctive operations to improve query understanding and disambiguate terms both in queries and documents (Baeza-Yates, 2004). There are many natural extensions that can be employed for such purposes e.g. synonym and query expansion, detection of phrases, lemmatisation, etc. Evaluation of the model will not rely heavily on the correct implementation of such and other linguistic processing techniques. In fact, the proposed model primarily perpetuates control of the retrieval process after these extensions have taken effect. The focus of this work is not about mapping synonyms or expanding query terms. The system can be improved with these extensions, but the primary focus remains on queries that accept well-defined concepts in a knowledge base.

The goal of the evaluation is to demonstrate the performance and usefulness of the model based on its novel aspects (besides its generalisation ability): its ability to propagate query evidence across entities, weigh and demarcate properties based on unsupervised learning heuristics, and expressive query modelling via external implicit or explicit parameterisation. The latter (expressive query modelling) is perhaps the most differentiating feature of the model and will occupy much of the evaluation, as it has

been subject to the least exploration in the literature.

The research questions to be answered revolve around whether the model brings together appropriate features and processes to reason effectively with a realistic Semantic Web data collection, and whether the presence of additional semantics in queries can improve the results (hence the need for more expressive query modelling). The research hypotheses discussed here are:

Hypothesis 1: The model will demonstrate effective performance on a selected dataset over a range of queries of variable complexity.

Hypothesis 2: The mixture of facts with text in queries (expressive, semi-structured queries) will significantly improve retrieval performance when compared to a similar set of simpler plain-keyword queries.

In both cases, the focus is on fully automatic query processing. In order to assess the two hypotheses, a mixture of statistical tests with qualitative analyses will be employed to compare different implementations of the model. Variations of standard Precision/Recall measures will be used to guide our decisions on *Hypothesis 1* (whether the model is indeed effective). Since there is currently no baseline system to compare against, the decisions on Hypothesis 1 will be mostly of qualitative nature (i.e. will not involve statistical significance tests). *Hypothesis 2* forms the basis of the evaluation and has been the decisive factor on the choice of dataset and queries to use for evaluation.

In order to assess Hypothesis 2, we will compare results (and evaluate their statistical difference) from two separate query sets of relative complexity crafted to represent the same information needs over the same dataset. The goal is to determine whether expressive queries with additional association semantics lead to significant improvements in performance over a similar set of simpler plain keyword queries (or the opposite – whether expressive queries end up degrading performance)[1]. For example, considering a

[1]This is an important topic to highlight: via comparison of simple versus expressive queries we will

catalog of government data records, Hypothesis 2 focuses on whether a query such as "datasets published by the UK Home Office" will lead to significantly superior results when compared to a query for "UK Home Office datasets" or simply "UK Home Office" (or queries involving multiple relations e.g. "datasets published by the UK Home Office and released in 2009"). A comparison between two sets of such queries will indicate (1) the model's ability to reason effectively with such additional restrictions (implicit restrictions) in queries, (2) whether there is any significant added-value from exploring a baseline level of semantics in a retrieval model.

The evaluation will also indicate whether the model can perform well with both types of queries (plain keyword and more expressive) or whether it is specialised to a particular style. Details on the evaluation methodology and the choice of measures employed are presented in the following section.

6.3 Evaluation Methodology

6.3.1 Selecting an evaluation dataset

Evaluation of ad hoc IR techniques is typically carried out in standard ways that enable uniform analysis of individual systems and comparison of different systems. To measure the effectiveness of techniques in a standard way, a test collection is required consisted of three things:

1. A document collection
2. A representative set of topics or queries to evaluate
3. A set of relevance judgements or assessments for each document-query pair

also be demonstrating that with little effort on the user's end to provide additional restrictions (with or without the support of simple interface features e.g. auto-completion or hint recommendation widgets) the outcome will be more effective with greater utility for the end user. The challenge is to construct separate query sets that will address the same information need, in standard (keyword) or expressive queries (semi-structured natural language based queries).

Document collections are expected to be representative for the application of the IR system. Queries should also reflect the types of queries expected of actual users of the application. Relevance judgements act as the golden-standard result and are usually provided by humans, through a process of pooling, classifying and filtering results from multiple search systems (this is particularly the case with very large document collections) (Manning et al., 2008; Croft et al., 2009). The goal of evaluation under these settings is to measure how well the ranking produced by an IR system corresponds to a ranking based on the user-defined relevance judgements. Relevance assessments should therefore be as comprehensive as possible, covering a large portion of the collection of documents; it is not uncommon for a new retrieval system to end up introducing new relevant items to the collection, a case that can lead to imprecise measurements (a result may be judged as non-relevant while it is in fact relevant, simply not in the relevant set).

Developers of conventional IR systems typically use available test collections provided by the main IR evaluation forums, such as the Text REtrieval Conference (TREC) and the Initiative for the Evaluation of XML retrieval (INEX), to demonstrate and evaluate the effectiveness of their models. These have proven exceedingly useful for cross-comparing systems and determining whether marginal improvements in Precision and Recall are worth the investment. Many of the collections used at the TREC conference series, such as GOV2 and the main Text Research Collection (5 volumes of various sub-collections), have been tested and improved over the years and constitute the most comprehensive sources available for text retrieval experiments.

In the case of Semantic Web experiments, the only available reference collection for entity-oriented searches is the Semantic Search Challenge dataset crafted for the 2010/11 semantic search competitions (Halpin et al., 2010). The collection utilises the Billion Triples Challenge 2009 dataset and provides a set of 100 keyword queries with associated relevance assessments. The collection is certainly a vital choice for enabling baseline comparisons with other competitors and is particularly useful for implementing and

evaluating indexing schemes for large-scale experiments (the raw triples alone constitute approximately 250GBs of data, which is a sizeable collection for proper deployment experiments).

It is questionable, however, whether the Semantic Search Challenge collection can be used to guide a discussion and analysis of our system's internals. Simply cross-comparing with other competitors will tell us very little about our system's internal capabilities and functioning, since the systems evaluated at the Challenge in both years 2010/11 attempted to combine various sets of features in their models (e.g. prior PageRank analysis of entities with query-dependent scores). It is unclear, therefore, which parts of the evaluated systems have contributed most to their ranks in the two competition runs, and chances are we will be comparing incomparable aspects of our model with the other contestants. In addition, the relevance judgements collected for evaluation appear to be biased towards selected domains, such as `dbpedia.org`, with 56% of the assessments in 2010 and 48% in 2011 made of DBPedia entries. It has been noted that systems have exploited the distribution of assessments to rank higher in the competition runs (Halpin et al., 2010). To this end, and in view of in-exhaustive relevance judgements, our system may produce more accurate lists yet fair worse than systems that focus on the popular domains.

For these reasons − in-exhaustive relevance judgements and in order to emphasise our own research hypotheses − we have chosen to use a more well-defined dataset and craft our own set of queries to evaluate the prescribed model. The types of queries developed for evaluation have a crisp SPARQL equivalence over the dataset, therefore the programmatic SPARQL results can act as the golden-standard result set that we need for Precision/Recall evaluation. In this way, we eliminate all external factors (in-exhaustive relevance judgements) that may cause the model to behave unexpectedly and can focus on developing the appropriate queries to emphasise our research hypotheses. We can then focus on tuning the model's parameters to reach a satisfying performance, without

having to worry about external noise. Additionally, by using a SPARQL query engine for a benchmark, we avoid a lengthy and costly process of collecting human relevance judgements. There are, however, a number of issues with using SPARQL results alone for the assessments, and these will be outlined in the concluding section of this chapter (Section 6.6).

6.3.2 Dataset overview

The dataset selected for evaluation involves a collection of Public Sector Information (PSI) catalogue records that has been harvested and used previously as part of an investigation for a global PSI aggregator service. This is the same dataset referred to in the example use-case covered in Chapter 1 - Section 1.3.1. A related publication is available with general information on the scope of these investigations (Koumenides et al., 2010). As mentioned already, the data comes from three government portals, namely `data.gov` (US catalogue), `data.gov.uk` (UK catalogue) and `data.australia.gov.au` (Australian catalogue), which have purposed their PSI catalogues online in HTML/RDFa or CSV formats. The catalogues were crawled from the respective portals, refurbished into formal RDF triples using a local domain/namespace and republished on the EnAKTing website in 2010 (http://catalogues.psi.enakting.org). The site provides various visualisations of the records as well (including a graph-based records correlator using the Protovis[2] Javascript library). Appendix B provides links to an online dump of the data.

The three refurbished catalogues constitute an interesting source of rich metadata embedded in a multitude of first-class objects. For example, departments, tags and agencies associated with the records were cleansed and projected onto real-world URIs. Other content was projected onto first class objects depending on the clarity of the original data. In some cases, it was necessary to normalise data, such as release dates, to a more uniform encoding standard e.g. ISO8601. We used in-house scripts to cleanse

[2]Protovis: `http://mbostock.github.com/protovis/`

dates so that they could be queried and visualised on the Web. Overall, the translation was intentionally kept to a minimum to avoid over-cleansing the original data.

Due to the heterogeneity of the data, the process resulted into three separate schemata for the three catalogues. Diagrams of the schemata and namespaces used for each catalogue are presented in Appendix B. There are noticeable differences between the three ontologies - the most striking example being the modelling of government bodies associated with the records:

– The US and AU catalogues contained a single definition for the publishers of datasets, which were modelled as subclasses of Dublin Core Agency in the respective transposed catalogues. In the case of the UK catalogue, there are three separate definitions for government bodies associated with its records. The original data (provided in CSV format) does not make a clear distinction of the type of association, other than the names provided in the column headers of the CSV file. These have been projected accordingly onto three separate entity definitions: Department, Author and Agency. There are cases where the same government body appears to be instantiated under more than one type, which introduces some noise in the data - a case that will be accounted for when analysing results in forthcoming sections. We have not gone through the extra effort of projecting the UK agency definitions onto a single class definition, since we would like to keep the data as close as possible to its original structure.

6.3.2.1 Summary statistics

To test the effectiveness of the model, a single resource network was built for the three catalogues. Summary statistics for the dataset are presented in Table 6.1. Overall, there are $41,998$ entities defined in the dataset (excluding 41 object and datatype properties), among which $5,997$ are catalogue records ($\sim$14.3$\%$). This results in a challenging task for the retrieval process, since most of the queries crafted for evaluation (next section)

Total triples	209,000
Total entities	41,998
Total record entities	5,997
Unique terms (unprocessed)	19,804
Unique stems	16,447
Maximum word frequency	16,793 (and)
Maximum stem frequency	7,870 (data)
Total word occurrences	508,855
Total stem occurrences	420,501
Words occurring $>$ 500 times	138
Stems occurring $>$ 500 times	130

Table 6.1: Statistics of the catalogues dataset.

involve picking out record URIs from the dataset. The task brings an interesting challenge when in the pool of resources only a minority few depict actual record entities.

Table 6.2 shows the 40 most frequent words extracted from the literal values in the collection, together with their frequencies (excluding any words that appear only in property labels or property URIs). Table 6.3 shows a similar list of 10 random lowest-frequency words from the collection. These are the lower-cased original terms prior to any stemming operations and elimination from a stop-list. A short list of stop-words has been applied in the experiments prior to any stemming operations to filter out very common English terms in the collection, as these would possibly yield low significance on the rankings and introduce additional noise in the dataset. It is important to note that common words referring to abbreviations were carefully picked out of the stop-list e.g. the common term “who” - standing for World Health Organisation.

6.3.3 Topics and relevance assessments

To demonstrate the effectiveness of the model over the catalogues dataset, we have manually crafted a set of 50 topics, or information needs, that have a crisp SPARQL

Word	*Freq.*	*Predicate*	*Word*	*Freq.*	*Predicate*
and	16,793	* (10859)	coverage	2,242	* (2234)
the	13,945	* (12371)	or	2,207	* (1799)
of	13,518	* (10599)	english	2,057	* (2020)
data	7,870	* (4696)	is	2,057	* (1884)
id	6,778	** (6610)	title	2,049	* (2043)
hub	6,531	** (6526)	language	2,015	* (1993)
for	6,102	* (4498)	alternative	1,966	* (1957)
statistics	5,491	* (4518)	designation	1,960	* (1960)
in	5,438	* (4262)	from	1,948	* (1665)
to	5,339	* (4078)	office	1,919	* (1803)
national	4,921	* (4092)	local	1,870	* (1522)
download	4,227	** (4207)	are	1,854	* (1794)
by	4,079	* (2846)	government	1,815	* (1642)
urls	4,064	** (4062)	catalog	1,799	*** (1779)
source	3,305	* (3270)	administrative	1,775	**** (965)
on	3,014	* (2719)	authority	1,771	* (1526)
as	2,830	* (2227)	health	1,609	* (1126)
england	2,439	* (1678)	year	1,590	* (789)
agency	2,358	* (2248)	time	1,568	* (1377)
information	2,350	* (1847)	this	1,563	* (1544)

* description ** label *** category-type **** specialized-data-category-designation

Table 6.2: Most frequent 40 words (lowercased) from the catalogues dataset. Third and last columns show the names of the properties/predicates that the terms are mostly associated with along with frequency information.

Word	*Freq.*	*Word*	*Freq.*
durum	1	nisra	1
haematology	1	encams	1
branciforte	1	spectrometry	1
cyrtandrae	1	hhs	1
multiflorum	1	distributive	1

Table 6.3: Low frequency words from the catalogues dataset.

equivalence over the dataset. The goal is to assess the model's ability to traverse interconnections between the records of the catalogues and arrive at the optimal result set, which will have a precise definition in the collection. Effectively, results from the model will be compared with the results produced from a SPARQL end-point (assuming the role of the gold-standard or ground-truth judgement of relevance).

A topic crafted for evaluation is made up of a natural language description (representing the actual information need) and a set of representative query formulas. An example topic is shown below:

```
{
    "num": 13,
    "description": "Find all datasets published by the Scottish
       Government and released in 2010.",
    "q1-simple": "Scottish Government 2010 datasets",
    "q2-expressive": "datasets released in 2010 and published by
       Scottish Government",
    "q3-sparql":
        "PREFIX ukd: <http://.../data.gov.uk/department/id/>
         PREFIX medp: <http://.../global/def/property/id/>
         PREFIX purl: <http://purl.org/dc/terms/>
         SELECT DISTINCT ?x
         WHERE { ?x <purl:publisher> <ukd:hc-245904526> .
         ?x <medp:date-released> ?o .
         FILTER regex(?o, '2010.*$') }"
}
```

The complete list of 50 topics crafted for evaluation is provided in Appendix C.1. There are two sets of keyword queries (`q1-simple` and `q2-expressive`) associated with the topics, as explained earlier, which will be used to compare the model's performance across two separately configured runs. The simple keyword queries (`q1-simple`) are made mostly of 1-5 keywords and contain no relations in them, such as "published by" or "part of". Simple keyword queries were cross-checked with the search engine logs of `data.gov.uk` to try and keep them similar to what users typically ask at such portals. Search engine logs were provided by the website administrator upon request in 2011. The more expressive queries (`q2-expressive`) are refinements of the simple keyword

queries to include additional semantics in them (mostly pertaining to properties, as in the example above). The idea is to demonstrate that simple association semantics in queries can improve results and that the system can reason effectively with them. The two keyword query sets are also of progressive complexity, and some of the more complex topics (e.g. topics 13 (example above), 23, 34, 36) include multiple constraints, such as publishers and release dates or catalogue categories.

6.3.4 Overview of evaluation process

The evaluation process is carried out using the TREC evaluation toolkit [3], a common open-source library with a comprehensive array of measures used for standard IR evaluations. The TREC toolkit accepts two input files, one containing the relevance assessments (commonly referred to as the "qrels file") and the other containing the output from the system ("results file").

6.3.4.1 TREC qrels file

In the qrels file, the fields are:

```
query-id iter doc-id rel
```

where `query-id` is the number of the query/topic, `doc-id` is the external identifier of the judged document (the URI of a relevant entity), `iter` is a constant set to 0 and generally ignored, and `rel` is the relevance assigned to the entity and can take on a series of values depending on the type of evaluation conducted. In our case, judgements are boolean and `iter` is always assigned the value of 1, indicating that the entity is relevant to the query.

6.3.4.2 TREC results file

In the results file, the fields are:

[3] TREC evaluation toolkit: `http://trec.nist.gov/trec_eval`

```
query-id iter doc-id rank score run-id
```

where `query-id` and `iter` are the same as before, `doc-id` is the external identifier of the retrieved entity, `score` is the score assigned to the entity by our system, and `run-id` is an identifier used for labelling the output of different runs. The `rank` field is generally ignored, as ranks are assigned internally by sorting on the `score` field.

6.3.4.3 System input/ouptut

The system accepts a free-form keyword query of relative complexity, a list of tuning parameters, and produces a ranked list of entity URIs accompanied by their individual scores. Parameter estimation is presented in the following section (Section 6.4.2). We do not place any constraints on the number of results to return, since queries have varying amounts of associated relevant entities. Results from each query type (simple and expressive) are aggregated into the appropriate results file and passed to the TREC evaluation toolkit to produce the output for the evaluation. The metrics selected for evaluation are outlined next.

6.3.5 Summary of evaluation metrics

The following measures have been selected to evaluate the performance of the model over each individual query set: Mean Average Precision (MAP), R-Precision, Reciprocal Rank, Precision at k documents, and Interpolated Recall-Precision Averages (these will also be used to construct Recall-Precision graphs to visualise the contrast of results from the different query sets). Appendix A offers an overview of each measure and what we expect to understand from the results.

Additionally, in order to assess (and indicate the significance of) the differences between the rankings produced by the two query sets (simple v. expressive) and evaluate Hypothesis 2, we check for statistical significance using *one-tailed* significance tests.

We have selected two comparison/hypothesis test measures to complement each other's outcome, namely, the *Wilcoxon signed-rank test* and *Student's t-test*. As with the rest, description and motivation for the choice of measures (along with what we expect to understand from the results) are provided in Appendix A.

6.4 System configuration

6.4.1 Text processing

Linguistic processing of terms has been constant across all the tests carried out in the experiments. Terms are filtered through a stop list, removed of any punctuation, and stemmed using the Porter stemmer[4] (Porter, 2006). An additional processing step has been applied to split **camel-case** words into individual terms e.g. the word "DirectGov" or "directGov" is split into the distinct terms "Direct" and "Gov". The same processing applied to terms in the collection dataset is also applied to terms in the queries.

6.4.2 Parameter configurations and settings

In order to instantiate the model operationally, several system parameters need to be specified and/or estimated. Weighting factors, such as term and entity dependencies, are extracted during the indexing phase and remain constant during query processing. Other parameters, specifically local object and datatype property dependencies, are provided during query processing and act as "knobs" in the inference to fine-tune the performance of the system. The conceptual specification of all conditional probabilities and other parameters has been the focus of the previous chapters.

[4]The Porter Stemming Algorithm: `http://tartarus.org/martin/PorterStemmer/`

Parameter	*Defined in...*	*Setting*
$W_{t_j,d_{i,u}}$	Sections 4.2.1 & 4.3.2.1, Equation 4.5	$1 + log(tf_{t_j,d_{i,u}})$
$W_{t_j,d_{i,u},q_1}$	Section 4.3.2.1, Equation 4.5	$1 + log\frac{N_{E,D_u}}{df_{t_j,E,D_u}}$
$p(E_i\|o_{j,k})$	Section 4.2.3, Equation 4.2	$\frac{p(E_i \wedge O_k)}{p(O_k)}$
γ	Section 3.5.2, Equation 3.6	0

Table 6.4: Constant parameters in all the tests.

6.4.2.1 Static/constant parameters

Constant parameters throughout all the tests in the evaluation are shown in Table 6.4. The choices of indexing weights (first two rows) are based on established measures from the literature. We use a variation of TF-IDF for term weights. $tf_{t_j,d_{i,u}}$ is defined as the raw frequency of term t_j in the value associated with entity E_i via the datatype property represented by $d_{i,u}$ (for the definition of local property variables refer to Sections 3.4.2 & 4.2.2). For the query term weights $W_{t_j,d_{i,u},q_1}$ we use an optimised IDF measure to account for property-level frequency information. N_{E,D_u} is defined as the number of entities that have property D_u in their local datatype contexts (i.e. entities associated with a local datatype property node descending from D_u). df_{t_j,E,D_u} is defined as the number of entities that are linked to term t_j via a local datatype property node descending from D_u (an instantiation of D_u).

The basic idea of IDF is that the terms that occur infrequently in the collection are more likely to be important, hence their presence in the query needs to be emphasised in the measurements. Use of IDF for query term weights effectively discriminates against very common terms in the collection. There is a some loss of information in our interpretation of $W_{t_j,d_{i,u},q_1}$ as IDF, since the measurements do not consider the type or

Variable-parameter settings – *simple query set*		
Strict consistency		Mixed Tuning
A1 (offset)	**B1** (optimal)	**C1** (optimal)
$p(d\|E) = 0.5$ $p(\neg d\|E) = 0.5$	$p(d\|E) = 0.9$ $p(\neg d\|E) = 0.1$	$p(d\|E) = 0.8$ $p(\neg d\|E) = 0.2$
$p(o\|E) = 0.5$ $p(\neg o\|E) = 0.5$	$p(o\|E) = 0.5$ $p(\neg o\|E) = 0.5$	$p(o\|E) = 7.0$ $p(\neg o\|E) = 7.0$

Table 6.5: Optimal (B1, C1) and offset (A1) variable-parameter configurations for the simple query set.

Variable-parameter settings – *expressive query set*		
Strict consistency		Mixed Tuning
A2 (offset)	**B2** (optimal)	**C2** (optimal)
$p(d\|E) = 0.5$ $p(\neg d\|E) = 0.5$	$p(d\|E) = 0.9$ $p(\neg d\|E) = 0.1$	$p(d\|E) = 0.9$ $p(\neg d\|E) = 0.1$
$p(o\|E) = 0.5$ $p(\neg o\|E) = 0.5$	$p(o\|E) = 0.6$ $p(\neg o\|E) = 0.4$	$p(o\|E) = 6.0$ $p(\neg o\|E) = 6.0$

Table 6.6: Optimal (B2, C2) and offset (A2) variable-parameter configurations for the expressive query set.

any other characteristic of individual entities.

The threshold γ on property matching has been set to 0, since most of the property definitions involved in the dataset and the queries are single-term labels. Setting γ to 0 allows global property nodes to be instantiated to $true$ when any one or more associated term nodes (parents of the global property) are found in the respective query layer.

6.4.2.2 Variable parameters – can be optimised and specified at query time

The remaining configurations involve the conditional probabilities of local object and datatype property nodes ($p(d_{i,k}|E_i)$ and $p(o_{i,k}|E_i)$ - Sections 3.4.2 & 4.2.2). These have turned out highly experimental estimates with their settings having a significant impact

on performance. A poor setting for the respective collection has been found to degrade performance by more than 100%. Optimal values are expected to vary across collections, while arriving at a global estimate would demand extensive experimentation with a potentially large number of datasets. We have not been able to verify any codependency between the two parameters either (essentially *four* parameters if we consider the states of properties), although there are some indications that we discuss later. For the current experiments, a Maximum Likelihood Estimation (MLE) measure was used to determine a set of ideal values for the two estimates to be used across all the queries. We describe our approach next.

In principle, values for the four parameters are expected to be strictly probabilistic, as outlined in Section 4.2.2, hence a single value for q should suffice for each of the two types of property dependencies. In practise, however, we have found that non-probabilistic independent values for the two states of the conditioned properties lead to superior results (sometimes close to 100% improvement). Tables 6.5 & 6.6 list the **offset** (in terms of property demarcation) and **optimal** (in terms of performance) configurations that we have arrived and will investigate to analyse and compare results for the two query sets. The tables show three settings for each query set, labelled accordingly (strict consistency v. mixed tuning) to indicate their correspondence. We discuss the implications of each next (Section 6.5), since these settings are the basis for our discussion and analysis.

As mentioned above, determining the optimal settings for the four parameters involved a MLE method. If we define D to represent a distribution of the system's output (we leave this abstract for now) and $\Theta = [\theta_1, ..., \theta_4]$ to denote a specification for the four parameters (dependencies $p(d_{i,k}|E_i)$, $p(\neg d_{i,k}|E_i)$, $p(o_{i,k}|E_i)$, $p(\neg o_{i,k}|E_i)$ resp.) then we are looking to maximise the function

$$\Theta_{MLE} = \underset{\theta_1,...,\theta_4}{\operatorname{argmax}} \; p(D|\theta_1, ..., \theta_4) \tag{6.1}$$

For the likelihood function $p(D|\theta_1, ..., \theta_4)$ we approximate using Mean Average Precision

(MAP), which is a probabilistic succinct summary of the system's performance and averages well over all queries. The task therefore involves finding a specification for the four parameters, denoted by Θ, that maximises MAP. The corresponding generalised distribution for the strict consistency settings requires an integral of the form:

$$\int_{0.5}^{1}\int_{0.5}^{1} p(D|\theta_1,\theta_3)\,d\theta_1\,d\theta_3 \tag{6.2}$$

since knowing the values of θ_1 and θ_3 we immediately have the corresponding estimates for the converse probabilities θ_2 and θ_4. For the mixed tuning configurations we have a more complex function to account for, and the corresponding generalised distribution:

$$\int_{0}^{\infty} p(D|\Theta)\,d\Theta = \int_{\theta_1}\ldots\int_{\theta_4} p(D|\theta_1,\ldots,\theta_4)\,d\theta_1\ldots d\theta_4 \tag{6.3}$$

spanning over all possible combinations of parameter values for $\theta_1,\ldots,\theta_4$. Although infinity is really the upper limit one could define values for the four parameters, the outcome will likely converge to a maximum at a very small integer/interval (in our case 6 and 7 for the two query sets), after which MAP will start to decay (although we have not verified whether performance may come back to a maximum at higher intervals). The process, therefore, need not necessarily check for every possible combination of parameter settings, but rather check for performance increases/decreases at sparse integers and, consequently, only integrate the function within a narrower range to reach an optimal. If the settings, however, are to be exposed for user fiddling/manipulation, then strict probabilistic intervals in the range $[0,1]$ would be the safest bet, although further experimentation may expose a more appropriate interval to embody the expected effects of the dependencies (Sections 3.4.2 & 4.2.2). Parameter interdependencies and probabilistic projections of all parameter values (regardless of scale) will need to be investigated in future research.

	Simple Query Set Baseline Results		
	Strict Consistency		Mixed Tuning
	A1	**B1**	**C1**
Queries Resolved	50	50	50
Retrieved Entities	366,662	366,662	366,662
Relevant Entities	20,346	20,346	20,346
Rel. & Retr.	20,292	20,292	20,292
MAP	0.3821	0.3995	**0.6130**
Rprec	0.4184	0.4277	0.6171
Reciprocal Rank	0.3847	0.3973	0.7488
P@5	0.3360	0.3520	0.6800
P@10	0.3540	0.3640	0.6280
P@15	0.3587	0.3627	0.5920
P@20	0.3530	0.3640	0.5740
P@30	0.3420	0.3607	0.5413
P@100	0.3010	0.3098	0.4456
P@200	0.2719	0.2749	0.4024
P@500	0.2182	0.2282	0.2695
P@1000	0.1811	0.1893	0.1874

Table 6.7: Summary evaluation results for the simple query set.

6.5 Evaluation Results and Analysis

6.5.1 Baseline results

Tables 6.7 & 6.8 show retrieval performance of the network model over the two respective query sets (simple and expressive queries). Results are provided for the different metrics and configuration settings discussed earlier, leading to variations in the rankings (please see Appendix A for a description of these metrics). Individual query results considering the best performing configurations (optimal) for all the queries in the collection are provided in Appendix C.2.

Configurations $A1$ and $A2$ are the same for both query types and have the effect

	Expressive Query Set Baseline Results		
	Strict Consistency		Mixed Tuning
	A2	**B2**	**C2**
Queries Resolved	50	50	50
Retrieved Entities	374,068	374,068	374,068
Relevant Entities	20,346	20,346	20,346
Rel. & Retr.	20,292	20,292	20,292
MAP	0.3641	0.4634	**0.7042**
Rprec	0.3965	0.4772	0.6772
Reciprocal Rank	0.3878	0.4613	0.8508
P@5	0.3120	0.4000	0.7640
P@10	0.3320	0.4120	0.7300
P@15	0.3453	0.4227	0.7027
P@20	0.3500	0.4410	0.6930
P@30	0.3473	0.4307	0.6627
P@100	0.2994	0.3522	0.5236
P@200	0.2358	0.2993	0.4542
P@500	0.1948	0.2542	0.3148
P@1000	0.1706	0.2057	0.2298

Table 6.8: Summary evaluation results for the expressive query set.

of neutralising the effects of property prediction in queries (hence the "offset" label in the table headings). This is the lowest performing configuration, particularly for the expressive query set.

Configurations $B1$ and $B2$ are the optimal settings observed when strict consistency to the inference formulas is desired. The variable parameter values remain strictly probabilistic, as outlined in Section 4.2.2. Average precision performance (MAP) improves 4.6% for the simple query set when configuration $B1$ is considered ($\%$ change from $A1$). Larger substantial gains are observed in the expressive query set (approximately 27.3% improvement from $A2$ in MAP when $B2$ is considered). The larger gains of $B2$ over $A2$ and the improvement over $B1$ (16% improvement in MAP) are indicative of the model's

ability to take advantage of the presence of associations in the more expressive queries. The parameter values of the $B2$ configuration are consistent with our expectations of the converse relationship between the states of properties, as outlined in Sections 3.4.2 & 4.2.2. Configuration $A2$ nullifies this effect and produces inferior results.

Configurations $C1$ and $C2$ are the ideal parameters for superior performance of the model over all the tests observed in the respective dataset. Parameter values involve a mixture of probabilistic and non-probabilistic estimates. Substantial performance gains over the optimal strict consistency configurations $B1$ and $B2$ are observed for all the evaluation metrics when $C1$ and $C2$ are considered. These are as high as 53.4% improvement for the simple query set and 52% improvement for the expressive query set. Considering the performance improvement from the baseline ("offset") configurations ($A1$ and $A2$), the mixed tuning configurations yield 60.4% and 93.4% (!) improvement for the two query sets, respectively. These differences indicate that proper settings for the parameters can have an astonishing effect on the model's performance. As indicated in Tables 6.5 & 6.6, local datatype property dependencies are always optimal when strictly consistent estimates are used (codependent probabilistic values). This has been the case in all the configurations investigated during our experiments.

A possible explanation for the high performance gains when non-probabilistic values are used for the local object property dependencies is the low diagnostic values accumulated at the entities from their local datatype contexts (inference formula 4.6). When these are filtered through the rest of the network via the local object contexts of other entities, their impact is diminished, sometimes to the point of very insignificant gains. When the initial probabilities are very low, accentuating the values via non-probabilistic dependencies (greater than 1) increases their impact on connected entities and performance is improved. A vital issue to consider when fiddling with non-probabilistic values is to avoid inflating the inference formulas with values greater than 1 at the point the disjunction operators take effect, as this would render the model inconsistent, with possibly

adverse effects on its performance. The current settings in $C1$ and $C2$ avoid this from happening, but a different dataset and a different query set may require readjusting these parameters.

Average precision at the rate of 70% is a promising outcome, while reciprocal rank in the range of $80 - 90\%$ means that the user will be seeing on average a relevant entity in the first result returned for each query. The decreasing precision values at higher ranks is partly due to the variation in the outcome space, or relevant results, for each query. Some queries have as low as 3 relevant entities, while others have as high as $4,149$.

The large number of retrieved entities is due to allowing the model to return all possible outcomes for every query. Precision estimates, however, are obtained after each relevant entity is retrieved, and the model proves capable of fetching relevant entities upfront. Average precision rewards systems that fetch relevant documents/entities quickly (highly ranked) and severely penalises otherwise. A threshold could have been placed to retrieve entities only above a certain accumulated belief, primarily to reduce additional processing from rendering all possible results in a production system.

The model proves particularly effective with 100% precision on a range of complex queries e.g. queries $5, 6, 13, 21, 23, 24$ and many others. When dealing with queries involving multiple criteria, the model accumulates diagnostic messages from multiple sources (datatype and object properies) into a disjunction operator, which allows it to increase the beliefs in entities, hence the indicative performance.

Sometimes our choices for text processing have an adverse effect on the ranks. For example, considering query 14 ("datasets published by DirectGov"), the model fails to retrieve any relevant entities prior to rank 500. First off, there are very few relevant results associated with the query (5 entities), which makes things particularly challenging for the model to reason. However, the term "DirectGov" itself is only associated 24 times to various entities in the data, via very common properties (properties pertaining to `rdfs:label` and `dc:description`). In the inference, this leads to two evaluations

of the IDF formula (the second variable in Table 6.4) to be factored into the cosine rule for evaluating the impact of query terms on the Resource Network (see Section 5.2.2.2 for a summary of evaluating $q1$ in the query). This results to significant IDF values in both cases, since there is a large difference between the number of times the two properties appear in the dataset as a whole and the number of times they appear as linked to "DirectGov". In effect, the initial weight given to the term "DirectGov" when present in the query is high enough to have a strong initial impact on the relevance of the entities linked to it. However, when camel-case processing is applied, the term "DirectGov", while an initially rare term in the collection, is split into the individual terms "direct" and "gov", which are both common terms in the collection. There are now over 445 entities linked to them directly via their local datatype contexts, and more entities linked to those via their local object contexts. As a result, the initial weight/impact of the terms "direct" and "gov" in the query is now defined by a lower IDF value than the original term "DirectGov". This causes the model to reason rather ineffectively. The discriminating power of the original term's IDF (IDF discriminates against common terms) has been weakened in favour of wider coverage.

Another query failure is noticed on topic 8. This is a query for departments referenced in `data.gov.uk`. The model fails to find an inference path linking department entities to the catalog entity, hence fails to retrieve any relevant results. The reason here stems both from the lack of predictive reasoning with backlink information and possibly the restriction in the inference to allow propagation only via directly connected entities. For example, even if the main `data.gov.uk` catalog entity (see ontology diagram in Appendix B) contained links to all the departments referenced in the catalog, the model would still potentially fail, since propagation is restricted to only flow towards entities via outlinks specified in their local object and datatype contexts.

	Precision			
	Strict Consistency		Mixed Tuning	
Recall	**B1**	**B2**	**C1**	**C2**
0.0	0.6439	0.7193	0.8199	0.9109
0.1	0.5751	0.6388	0.7654	0.8711
0.2	0.5480	0.6061	0.7444	0.8477
0.3	0.5169	0.5692	0.7146	0.8085
0.4	0.5099	0.5610	0.6906	0.7822
0.5	0.4644	0.5192	0.6530	0.7245
0.6	0.4367	0.5055	0.6325	0.7127
0.7	0.4164	0.4887	0.6090	0.6774
0.8	0.3975	0.4585	0.5752	0.6460
0.9	0.3728	0.4181	0.5422	0.6158
1.0	0.3083	0.3441	0.4549	0.5214
average	0.4718	0.5299	0.6547	0.7380

Table 6.9: Interpolated recall-precision results for the best performing configurations for the two query sets (simple: [B1, C1], expressive: [B2, C2]).

6.5.2 Comparison and hypothesis testing

Table 6.9 shows retrieval performance of the model in terms of average precision at 11 standard recall levels. Values are provided for the two optimal configurations for each of the query sets. As previously, results are based over all the entities retrieved for each query. Figures 6.1 & 6.2 show the recall-precision values plotted on a recall-precision graph, cross-compared over the two query sets. Precision values are interpolated, hence the smooth curves and the monotonically decreasing function at increasing recall levels. Use of interpolation defines precision values at the recall level 0.0, which, for a representative system, would fall between 0.7 and 0.8 (Croft et al., 2009). Results from our best performing configurations are higher than average, found at the peak levels of 0.82 and 0.91.

Performance at the standard recall levels improves by 12.3% (B2) and 12.7% (C2)

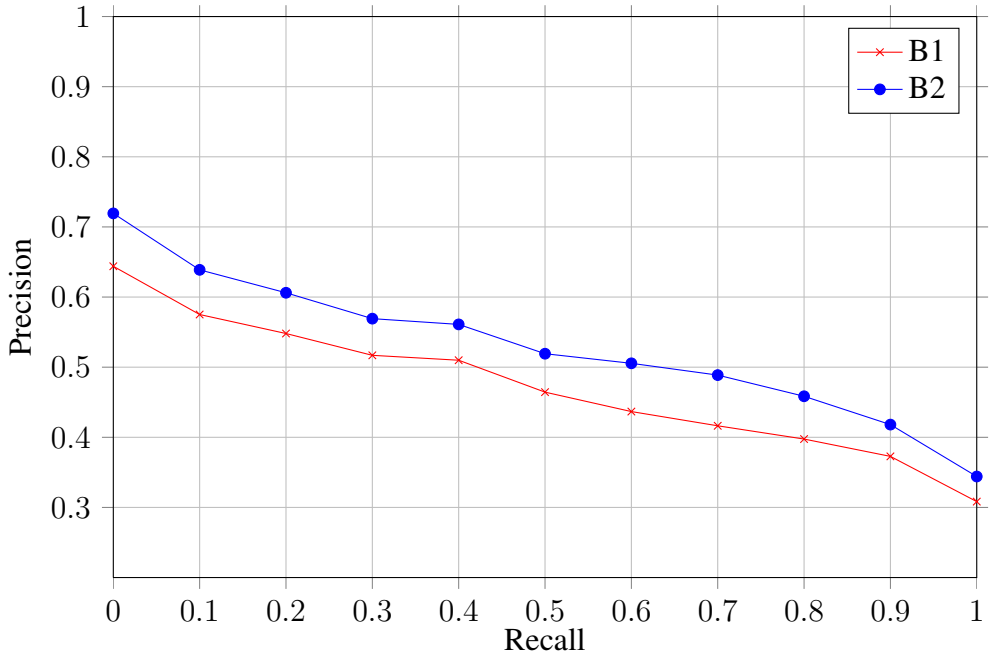


Figure 6.1: Interpolated precision-recall graph for the best performing "strict consistency"configurations for the two query sets (simple: B1, expressive: B2).

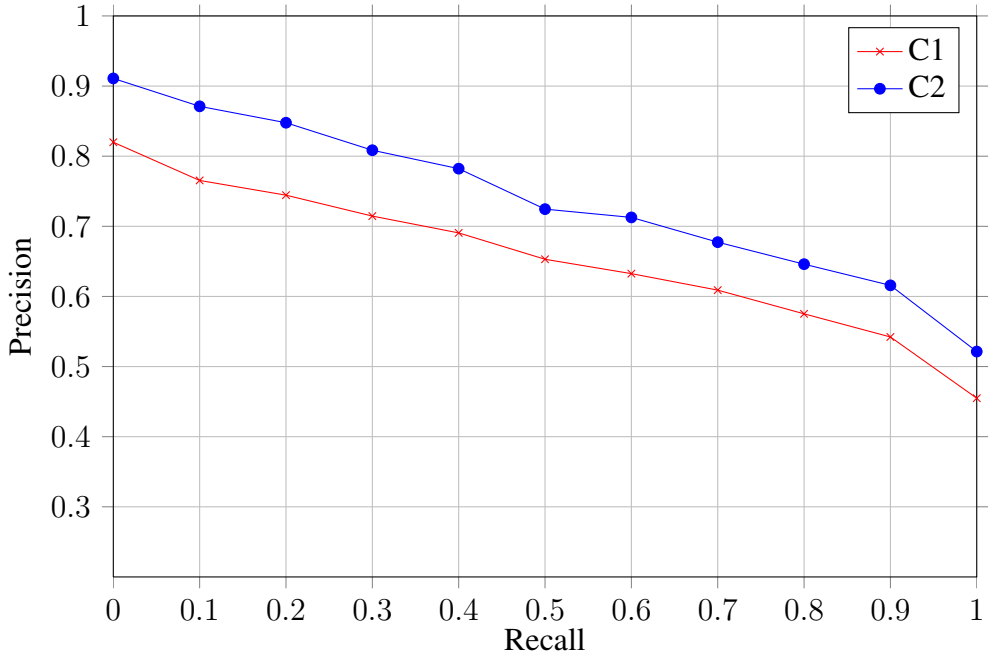


Figure 6.2: Interpolated precision-recall graph for the best performing "mixed tuning" configurations for the two query sets (simple: C1, expressive: C2).

when expressive queries are considered. This is also obvious from the interpolated precision curves. Results from the expressive queries (configurations B2 and C2) yield superior performance over all standard recall levels. Similar improvements are observed in all the aforementioned tests of the evaluation (Tables 6.7 & 6.8). Overall, the model yields superior performance on all counts when expressive queries are evaluated.

In order to indicate the significance of the differences between the two rankings and evaluate Hypothesis 2, we check for statistical significance using the *one-tailed* significance tests outlined in Appendix A. Results from the comparison of the highest performing configurations across the two query sets (configurations C1 & C2) are presented for the Wilcoxon test in Table 6.10 and the Student's t-test in Table 6.11. Explanations for these tests have been provided in Appendix A. For convenience, P-values are extracted for both one-tailed and two-tailed tests.

Considering Table 6.10, W is the sum of the signed ranks, or the observed value of the Wilcoxon test, $n_{s/r}$ is the number of queries that have non-zero differences in each of the measures compared across the two query sets (MAP, P@10, P@30, P@100, etc.), and the remaining columns are self-explanatory. In all of the Wilcoxon tests, we observe significant statistics at the 0.05 level ($Z > 1.645$), which indicate at least a 95% confidence in Hypothesis 2. In most of the tests, significance is observed at the 0.025 level ($Z > 1.960$) and in some tests even at the 0.01 level ($Z > 2.326$). These results lead us to accept Hypothesis 2. There is a statistically significant difference between the rankings produced from the optimal configurations of the model across the two query sets, which is unlikely to be produced by a random set of data (in place of either implementation). The model proves effective to reason with queries containing multiple associations, which are a reasonable explanation for the statistically significant superior results over the plain-keyword (less expressive) queries.

To substantiate the results from the Wilcoxon test, we attempt to re-affirm our hypothesis with similar tests using Student's t-test. Table 6.11 shows results from the

	Wilcoxon Signed-Rank Test Results				
	W	$n_{s/r}$	Z	P (1-tail)	P (2-tail)
MAP	342	41	2.21	0.0136	0.0271
P@10	99	19	1.98	0.0239	0.0477
P@30	226	27	2.71	0.0034	0.0067
P@100	194	26	2.46	0.0069	0.0139
P@500	86	16	2.21	0.0136	0.0271
P@1000	66	12	2.57	0.0051	0.0102
iPrec@0.1R	252	30	2.59	0.0048	0.0096
iPrec@0.3R	277	35	2.26	0.0119	0.0238
iPrec@0.7R	253	37	1.90	0.0287	0.0574

Table 6.10: Wilcoxon signed-rank test results based on the comparison of results from the best performing configurations for the two query sets (C1 and C2).

Student's t-test over the same measures used in the Wilcoxon test (MAP, P@10, P@30, P@100, etc.). Results are based on a *paired* t-test, which implies that each data point (evaluation result per query) in one group corresponds to a matching data point in the other group, for every measure assessed (i.e. scores are cross-matched on a per query basis). Hypothesis 2 is re-affirmed, perhaps even stronger, with the Student's t-test. All t-test results indicate significant differences in the results produced by the two implementations (configurations C1 & C2) at the 0.05 level, while most measurements indicate significance at the 0.025 and 0.001 levels (considering the one-tailed test results).

In conclusion, results from the aforementioned tests lead us to reject the Null hypotheses and accept Hypothesis 2: *The mixture of facts with text in queries (expressive, semi-structured queries) will significantly improve retrieval performance when compared to a similar set of simpler plain-keyword queries*. Given the evaluation settings, the model proves significantly more effective on average when additional restrictions are present in queries (implicit restrictions/associations/semantics). Results are *indicative* of this phenomenon but not *conclusive*. For a more conclusive answer and confirmation, we

	Student's t-test Results	
	P (1-tail)	P (2-tail)
MAP	0.00993	0.01986
P@10	0.02359	0.04719
P@30	0.00240	0.00481
P@100	0.01712	0.03424
P@500	0.02803	0.05606
P@1000	0.01514	0.03028
iPrec@0.1R	0.00387	0.00773
iPrec@0.3R	0.00864	0.01727
iPrec@0.7R	0.01928	0.03857

Table 6.11: Student's t-test results based on the comparison of results from the best performing configurations for the two query sets (C1 and C2).

would need to assess the model over multiple sets of data and queries, which can be the main line of future research.

6.6 General Discussion

This chapter presented an evaluation of the model over a realistic data collection, on par with the use case explored in Chapter 1. The evaluation focused on comparing the performance of the model over a range of queries against a SPARQL end-point with an equivalent set of SPARQL queries. The SPARQL results served as an optimum benchmark, a "gold standard" result set, on which to contrast the performance of the model. As expected, performance is lower than that of a SPARQL end-point. The model, however, is intended for free-form text search, not a substitute to SPARQL. The experiments have illustrated that the model offers several parameters to fine-tune its performance over particular collections and types of queries. Some parameters can be fiddled at query time, which can lead to improved results. The focus of the evaluation,

however, was on average performance.

The results of the evaluation indicate promising performance and can serve as a baseline for evaluating future extensions to the model. This is an important aspect of the evaluation. We have illustrated how to tune the model (although, as discussed in the following chapter, additional tuning is possible), outlined all the variable and fixed parameters, and divided the evaluation across two query sets that expose particular functionality in the model. In the following chapter, we discuss ways to enhance the model to potentially improve its performance. As one of our foremost criteria, we have developed an extensible model that can serve as the backbone for future experiments. The evaluation results provided here can serve as the baseline for evaluating such additional operations on the model.

There are a number of difficulties and drawbacks, however, in using a strictly programmatic approach (SPARQL queries) to serve as the benchmark for evaluation. The arbitrariness and sometimes unclear nature of the semantics in the data also lead to significant challenges when configuring query sets and respective relevance assessments. These have had an impact on the performance of the model and the choice of queries used for evaluation. In the remaining of this section, we discuss issues regarding the prescribed method of evaluation, data, and queries.

6.6.1 Choice of queries

There is a very limited set of keyword queries that can be resolved deterministically with SPARQL equivalents. For example, queries such as “datasets about crime” or simply “crime” have a very ambiguous definition in the knowledge base. Foremost, the term “crime” may appear as part of a longer multi-word tag, in the description of the dataset, or even in the name of the dataset. Furthermore, a synonym or other associated term may be used instead of the actual word “crime” to depict certain records as being about crime e.g. the word “burglary” or other synonymous term. Attempting to resolve such queries

strictly via REGEX constructs in SPARQL queries will lead to several false positives (Manning et al., 2008) in the result set, which would need to be filtered out by several human judges to arrive at a precise ground-truth answer set. For these reasons, we have chosen to keep a strict focus on queries that have a crisp correspondence to SPARQL equivalents.

6.6.2 Unclear relevance

In a similar manner to the aforementioned, it is unclear whether the URI of a tag with the label "crime" is relevant to a query for "crime datasets" e.g. a user may or may not be satisfied with retrieving a tag URI that links to all the records tagged with crime. This, in fact, applies to almost all the queries formulated for the evaluation (whether related to agencies, categories, or the catalog itself). For example, it is unclear whether the URI of an agency is relevant to queries involving records associated with that agency e.g. a user may or may not be satisfied with retrieving the URI of an agency that links to all the records published by the respective agency. We have chosen not to include these additional URIs in the gold-standard result sets, which may have had implications on the performance of the model, particularly on queries with very few expected results.

6.6.3 Unclear semantics and false negatives

A final issue involves some of the topics selected for evaluation that may result in a bigger pool of relevant answers (false negatives (Manning et al., 2008)) when resolved by our model. This is the case with records in the UK catalog, where the same government institution acts as either the agency, department or both in several dataset definitions. For example, DirectGov appears to be the agent of two datasets and the publisher of five others. Considering a query for "datasets published by DirectGov", the SPARQL results assume that the user is precisely looking for the five datasets whose publisher is

DirectGov. However, the other two datasets could very well be relevant. Would a user not be interested in datasets where DirectGov is only the agent but not the publisher of the dataset? Does the user know the distinction between an agency and a department? Is there any pragmatic difference between the two associations (the properties "publisher" and "agency")? In such cases, where semantics are unclear in the dataset, we expect to witness noise and unpredictability in the ideal (SPARQL) result set. As mentioned previously, we have not cleansed the dataset to account for this type of noise in the data, as it is not uncommon to witness such ambiguous semantics in real-world settings, especially with datasets provided via uncontrolled dissemination processes (a common case with several LOD collections).

For some of the topics assessed, we formulate two SPARQL equivalents to account for both cases, although the results used for measuring average performance of the model are strictly be of the former case (where the distinction between the properties "publisher" and "agency" is expected to be clearly understood − hence more challenging for the model).

Chapter 7

Conclusions and Future Research

Semantic search is an exciting area of research that brings together topics from Information Retrieval, the Semantic Web and other areas where semantics are prevalent in the modelling and representation of information resources. The rise of standards for semi-structured machine processable information and the increasing awareness of the potential benefits of a semantic Web are leading the way towards a more meaningful Web of data, which in itself is being realised through a literally exponential increase of published data and continuous refinement of standards and production of application systems. Questions regarding location and retrieval of relevant data remain fundamental in achieving a good integration of disparate resources and the effective delivery of data items to the needs of particular applications and users. We consider the basis of such a framework as an Information Retrieval system that can cope with semi-structured data. At the same time, exploiting the availability of a semantic structure opens a plethora of possibilities for enhancing and complementing traditional search. Our understanding of information resources is enhanced through a finer granularity at the level of objects, while conventional interaction mediums can be enriched via a semantic structure to potentially enhance their effectiveness and utility for end users.

There are many convoluting factors to consider when addressing information access at the semantic level. Semantic search is a dynamic area of research with many areas

of specialisation. In the mix of practise, we find research focusing on a number of spectrums e.g. systems that exploit the semantic structure to enhance user interaction via graphical or iterative information access patterns, Machine Learning techniques for reasoning with graph-based data via conventional or semi-structured information requests, or light-weight retrieval models and indexing schemes for efficient search across multiple heterogeneous data sources. As with conventional Information Retrieval research, the fundamental questions to be answered lie within the context of the task being prioritised.

In this thesis, we advocate the development of a semantic Information Retrieval model that is built from the ground up with semantics in scope. The model aims to facilitate access to data by enabling reasoning with natural language based queries of relative complexity, with query semantics specified explicitly by users or incorporated via more implicit bindings. The model is based on Bayesian Networks and, as customary with similar approaches in Information Retrieval, tries to generalise into a single computational framework the necessary constructs to reason with several sources of available knowledge. The model differs from similar deployments of Bayesian Networks in Information Retrieval in that it aims to represent, and make explicit in the inference process, the presence of multiple relations that potentially link semantic resources together or with primitive data values, as it is customary with Semantic Web data. To this end, the thesis seeks to contribute to a better understanding of the use of Bayesian inference networks to support entity search in semantic knowledge bases. The ground foundations of the model offer a rich setting to satisfy an interesting set of queries and incorporate a variety of techniques for fusing probabilistic evidence, both new and familiar.

7.1 Summary of the Thesis

In this thesis, we have introduced the area of semantic search from a broad point of view and subsequently narrowed our focus to key techniques from the literature (focusing on ranking methods and auxiliary processes) involving keyword and semi-structured search over Semantic Web data (Chapters 1 & 2). Our coverage maintains a focus that is inline with the scope of recent workshops, competitions and conventional Information Retrieval conferences focusing on entity-oriented searches.

Subsequently, in Chapters 3 & 4, we introduced a new retrieval model for reasoning with keyword and semi-structured natural language queries in Semantic Web knowledge bases. We achieved this by developing a generative expressive Bayesian Network model that is capable to express the explicit semantics associated with resources and expose them to statistical scrutiny and generation of inference procedures. We employed a variety of techniques to leverage the available semantics in the data (mostly focusing on interrelations between data items) to bring together a unified ranking procedure with a sound mathematical foundation and potential for further extensions and modifications.

In Chapter 6 of the thesis, we concentrated on evaluating the model to offer an overview of what it delivers when applied over a realistic data collection and to indicate its demonstrated performance. The focus of the evaluation was on the effectiveness of the model to reason and respond appropriately to the key retrieval features and aspects presented in previous chapters (Chapters 3 & 4). Besides its generalisation ability, the distinctive aspects of the model are: its ability to propagate query evidence across entities, weigh and demarcate properties based on unsupervised learning heuristics, and expressive query modelling via external implicit or explicit parameterisation. The latter (expressive query modelling) is perhaps the most differentiating feature of the model and thus occupied much of the evaluation, as it has been subject to the least exploration in the literature. In order to prepare an evaluation methodology, we developed a set of research hypotheses to be accepted or rejected according to statistical and qualitative

analyses of the model's performance.

For the evaluation process, we selected a previously harvested dataset of government catalogue records and crafted manually a set of 50 evaluation topics, made of two separate sets of queries aimed to emphasise our research hypotheses. The two sets of queries involve different specifications of the same information needs (evaluation topics) using either plain keywords or more expressive queries (involving a mixture of facts with text/ implicit conditions/relations). One of our goals in the evaluation was to determine whether expressive queries with additional association semantics lead to significant improvements in performance over a similar set of simpler plain keyword queries (or the opposite $-$ whether expressive queries end up degrading performance). The topics crafted for evaluation had a crisp SPARQL correspondence over the dataset, hence the programmatic SPARQL results acted as the golden-standard result set that we needed for Precision/Recall evaluation. The evaluation methodology utilised standard evaluation metrics employed conventionally in IR experiments and statistical significance tests to compare different configurations of the model over the different query sets.

The model was suitably evaluated and demonstrated promising performance over the collection of queries and data selected for evaluation. Results indicated that the model utilises properly the additional semantics in queries to propagate and demarcate evidence in the network, achieving promising performance over a range of expressive queries. The model proved capable with similarly strong performance to reason with plain keyword queries. In its optimal configuration over the expressive query set, the model achieved Mean Average Precision at the rate of $\sim 70.4\%$, which is a promising outcome for a baseline performance estimate. Similar results had also been observed from the remaining metrics used in the evaluation. Expressive queries yielded statistically significant superior results when compared to results from the plain-keyword queries, which is evident of the model's ability to reason with various types of queries of relative complexity.

Worth mentioning is that some of the queries crafted for evaluation were particularly challenging to resolve and the dataset involved a range of ambiguous contextual information and a variety of properties and entity types. Record entities alone (the main targeted type) constituted only $\sim 14.3\%$ (approx. $6,000$ entities) of total entities in the dataset, with some queries requiring less than 10 entities to be resolved. Overall, the model's performance is indicative of a capable model to reason effectively with a range of queries of relative complexity. As mentioned elsewhere in the thesis, results are *indicative* of this phenomenon but not *conclusive*. For a more conclusive answer and confirmation, we would need to train and assess the model over multiple sets of data and queries, which can be the main line of future research.

7.2 Directions for Future Research

The Information Retrieval model developed in this work offers a framework of promising performance to achieve text-based search over triple/graph-based knowledge bases. The model, however, is presented as an initial engagement on which we can extend with various additional functionalities, enhancements and efficiency improvements, and alignment with a more general complex of tools for keyword querying in knowledge bases. In the following, we outline key areas that future research may be directed to extend the model and utilise it in proper production contexts.

7.2.1 Further training and parameter tuning

As with any Information Retrieval or Machine Learning experiment that depends on a set of adjustable parameters to produce a desired outcome, our model encompasses a set of tuning factors that can be fiddled to fine-tune its performance and carry it through to other application contexts. We have presented a potential use case in the thesis as part of a government locator service and used a realistic data collection to train the model and

demonstrate its indicative performance over the respective dataset. The model proved capable to reason with a wide range of queries, while substantial performance gains were witnessed after properly tuning the system parameters.

We have presented two variational models in the thesis configured on different parameter ranges: one that was strictly consistent to the theoretical constraints and underpinnings of the inference formulas, and one that violated the consistency by introducing non-probabilistic values in the inference. The mixed configuration settings led to superior results in the rankings, sometimes close to a 50% gain in performance. Our results indicate that tuning the model parameters (particularly when a fully automatic system is desired) can lead to promising and perhaps even surprising outcomes. The focus of the evaluation was on average performance, hence parameter settings depict general configurations for the model. It is reasonable to expect even better performance had the parameters been tuned on a per-query basis, which is their intended use for supreme performance and proper external parameterisation.

Further training on additional collections (which can be constructed manually by following the guidelines in Chapter 6) to learn the convergence tendencies of the parameters would be a natural progression in the model's development. Moving on to macro settings to test the model over a larger knowledge base made of multiple datasets (preferably of different types of content e.g. statistical, biological and editorial) would be a natural next step in the process. Questions that demand further investigation include: (1) general convergence tendencies and interdependencies of the variable parameters in the model, (2) probabilistic projections of variable parameter values from a non-probabilistic scale, (3) the co-dependency of static and variable parameters (especially for the local object contexts of entities $-$ subject to two levels of property demarcation: via external parameterisation and unsupervised property weighting), and (4) the absolute maximum average performance achievable over a collection if parameters are tuned on a per-query basis.

parameter values (regardless of scale)

7.2.2 Reasoning extensions

The bulk of reasoning involved in the inference encompassing the model (Chapter 4) was diagnostic/likelihood processes emanating evidence via outlinks across connected entities. An attractive characteristic of the model is its generic network structure that enables various forms of additional reasoning, which can potentially improve performance in given contexts.

7.2.2.1 Backlink propagation

We have not investigated evidence propagation via backlinks associated with entities. However, as indicated during evaluation of the model, backlink information is essential to resolve certain types of queries (depending on how the data is originally modelled). The problem with embodying a straight out propagation process based on backlinks in a network is that we could end up losing or distorting part of the identity of resources due to an uncontrolled propagation process. If evidence is delivered to support the relevance of a given entity from any other entity that potentially links to it, then the outcome could end up distorting its true identity i.e. a person cannot control what is being conferred about him/her from surroundings entities. We had thus concentrated on the identity of resources as provided in their concise descriptions that link them via outlinks to information in the network.

Backlinks can be weighted and used in the same manner that outlinks were encoded. The difference is in the ranking formula, in which backlink propagation would be contributing to the prior or predictive support accorded to entities i.e. we would be introducing evidence in the marginal probability of entities $p(E_i|b)$, where b is the set of all entities linked to E_i via their local object contexts[1]. We could smooth this evidence

[1]The challenge therefore lies in quantifying, or subjecting, the conditional probability $p(E_i|b)$.

to lessen its impact on the final posterior via a convex combination, and adjust the smoothing factors according to which form of evidence (predictive v. diagnostic) we wish to emphasise. This is one idea of encoding backlink information.

7.2.2.2 Dataset information

We have presented studies in Chapter 2 (e.g. the Sindice (Oren et al., 2008) and SWSE (Hogan et al., 2011) experiments) that encode dataset information in the ranking process. The importance/popularity of a dataset (e.g. the pay-level domain of a resource) proved an interesting source of information and was captured via PageRank analysis using linksets connecting datasets together. There are many possibilities of encoding dataset information in our model. The easiest way would be to measure popularity scores for datasets, following the same or a similar procedure to the one outlined in (Delbru et al., 2010b), and introduce them as prior support to the entities. Another use of popularity scores would be to smooth the diagnostic messages emanating from query-activated entities to the local object contexts of other entities, thus encoding them directly in the inference formulas presented in Chapter 4. Since these scores would be computed a priori, they would not incur any costs on performance. The question is whether we trust popularity scores to alter the impact of evidence disseminated in the network.

7.2.2.3 Distant entity propagation

Allowing propagation to reach or originate from distant entities in the model raises the risk of double counting evidence, either sent from a given entity and delivered back via its local object context or the same message disseminated to multiple connected entities. This is due to the network being inherently multiply connected, and as more "hops" in the network are processed during the inference process the chances of double counting increase. There are, however, queries that cannot be resolved from considering only query evidence delivered from directly connected entities. Examples have been presented

during evaluation of the model and in the introductory notes of Chapter 2.

Propagation and inference in multiply connected Bayesian Networks is an issue that has received much attention historically in the Machine Learning and Artificial Intelligence literature. Generally, the case leads to approximated or assumption-based reasoning, since loops or cycles in the underlying network may cause messages to circulate indefinitely or towards an asymptotic equilibrium that does not reflect the true posteriors of resources.

In our case, we have only accounted for diagnostic messages in the inference, which incidentally gives us more control over the process. However, the case is still nontrivial. Even the slightest arithmetic approximation can have a significant impact on the rank of entities. A solution that would allow evidence propagation across distant entities would demand a strict and possibly resource-expensive process to determine how messages are accumulated in the inference formulas and which ones ought to be delivered back to connected entities.

7.2.3 Production release

In this thesis, we have chosen to focus on a single mode of user interaction and presented our findings on a novel algorithmic approach for distilling information from knowledge bases to satisfy user queries. From a broader perspective, however, semantic search is commonly viewed as an iterative and exploratory process in which the user can actively engage with the system via various forms of interaction (Uren et al., 2007; Hildebrand et al., 2007). The idea is to help the user explore the domain, find out what is there and construct complex queries from possibly several atomic or incremental operations.

An interesting direction for future research is how to manage the integration of end-user support utilities, such as multi-facet views, class menus and visualisation graphs, auto-complete functionality, and pre/post-query disambiguation components with our ranking heuristics to accomplish a more comprehensive and multimodal design model.

How these can feed back and affect the ranking or complement the model's ability for external parameterisation is an attractive prospect to investigate. Research into cognitive aspects will be important in this context, such as how much interaction a user is willing to bear to improve her search results. Once a suitable bundle of optional and/or customary search engine components are aligned with the core architecture of the model, then we can look forward to a standard library release. A mature, off-the-shelf component that can be adapted readily atop of existing knowledge base stores is an attractive prospect.

Appendix A

Summary of Evaluation Metrics

The measures used for evaluating the effectiveness of the model come from standard evaluation metrics employed conventionally in Information Retrieval experiments. See any standard textbook for more details on Recall/Precision evaluation (Croft et al., 2009; Baeza-Yates and Ribeiro-Neto, 1999; Manning et al., 2008). In the following, we offer a brief overview of the methods selected for Recall/Precision evaluation and statistical significance testing (model implementation comparisons).

(Mean) Average Precision

Average Precision is a popular method used for summarising the effectiveness of ranking produced by retrieval systems. The method is an average of the precision values (percent of retrieved documents that are relevant) from the rank positions where a relevant document is retrieved (i.e. when recall increases). The value depends heavily on the highly ranked relevant documents, and where a relevant document is not retrieved at a given rank, the contribution of that document to the average is 0.0. Average Precision is a non-interpolated measure. When multiple queries are considered, effectiveness is measured by averaging the individual Average Precision numbers to get a single number

for the performance of a system, known as Mean Average Precision (MAP). The MAP measure provides a very succinct summary of the effectiveness of a ranking algorithm over many queries.

R-Precision

R-Precision measures precision after R docs have been retrieved, where R is the total number of relevant documents for a query. If R is greater than the number of documents retrieved for a query, then the non-retrieved documents are all assumed to be non-relevant. The measure is averaged over all queries to produce a single summary number. Although the measure has a number of disadvantages (e.g. does not distinguish between different rankings of a given number of relevant documents), it works as a general approximation of a system's performance and is trivial to understand.

Reciprocal Rank

Reciprocal Rank is often used for applications where there is typically a single, or very few, relevant documents. Some of the queries explored in the experiments are of this nature, where relevant entities are 5 or less. The measure is defined as the reciprocal of the rank at which the first relevant document is retrieved. The measure is very sensitive to the rank position and can fall drastically as lower ranks are considered e.g. falls from 1.0 to 0.5 from rank 1 to 2. The average Reciprocal Rank is the average of the Reciprocal Ranks over a set of queries. This is a useful measure for indicating whether a system manages to produce relevant documents as the foremost answers to a query.

Precision at k documents

Precision after k documents have been retrieved is a useful method for determining how

many good results there are at fixed low levels of retrieved documents. The measure does not factor in recall. If X documents are not retrieved for a query, then all missing documents are assumed to be non-relevant. Values are averaged over all queries for summary evaluation. Precision at k is particularly useful for Web search applications, where what matters most is how many good results there are on the first few pages of search results (hence can emphasise on Precision at 10 or 20). The disadvantages are that the measure does not distinguish between different rankings of a given number of relevant documents and that it does not average well because it is strongly influenced by the total number of relevant documents for a query.

Interpolated Recall-Precision Averages

Interpolated Recall-Precision is used for summarising the effectiveness of ranking at fixed or standard recall levels (after a certain percentage of all the relevant documents for a query have been retrieved). The measure is otherwise known as the *eleven-point interpolated average precision* because it measures precision (percent of retrieved documents that are relevant) at the 11 recall levels of $0.0, 0.1, 0.2, ..., 1.0$. Since recall values are unlikely to be available at these levels to enable averaging across queries, precision is interpolated as the highest precision found at any recall level above the standard levels. For example, precision at recall 0.10 (i.e. after 10% of relevant documents for a query have been retrieved) is taken to be maximum of precision at all recall points ≥ 0.10. The justification for this type of interpolation is that users would be prepared to look at a few more documents if it were to increase the percentage of viewed documents that are relevant. The method has the advantage of summarising the effectiveness of the ranking of the entire set of relevant documents, rather than just those in the top ranks. Use of interpolation produces a monotonically decreasing function, reflecting the general notion of precision tending to decrease or stay the same as recall increases. Values are averaged

over all queries for each of the 11 recall levels and can be used to create Recall-Precision graphs (also useful visual comparisons of models and/or queries).

Statistical significance tests − for comparison

In order to assess (and indicate the significance of) the differences between the rankings produced by the two query sets (simple v. expressive − see Chapter 6 for details) and evaluate the *2nd* research hypothesis (Section 6.2), we check for statistical significance using *one-tailed* significance tests.

Statistical significance tests produce a *P-value* (or *Z score* from which a P-value can be approximated), which is the probability that a test statistic at least as extreme could be observed if the null hypothesis were true. The null hypothesis is a default hypothesis, which in our case (considering Hypothesis 2) would imply that there is no significant or meaningful difference between the ranks from the two query sets. The null hypothesis is often rejected in favour of the alternative hypothesis when the P-value is less than a significance level α, typically set to either 0.05 or 0.01 (indicating a 95% or 99% confidence level in the alternative hypothesis, respectively). The corresponding critical *Z* score for a 0.05 one-tailed significance level is ± 1.645 and ± 2.326 for the 0.01 level (when evaluating *Z* scores alone, we would want to obtain values not in the $\pm$ range of the critical values). The reason for α to be very small is to reduce the chance of a Type I or Type II error (incorrect rejection or acceptance of a null hypothesis, respectively). The following significance tests are used in our experiments to evaluate Hypothesis 2:

Wilcoxon signed-rank test. The Wilcoxon signed-rank test is a familiar measure used for comparing results from different retrieval systems. The test belongs to the family of non-parametric statistical tests and makes very few assumptions about the data and the underlying distribution. It is often used as an appropriate alternative to the parametric *t-test* when the sample data fails to meet the necessary assumptions,

such as random sampling from a standard normal distribution. The null hypothesis for a Wilcoxon signed-rank test is that the *median* difference between the pairs of observations is zero. We compute the test statistic and extract the appropriate Z scores for the differences in Average Precision, P@{10, 30, 100, 500, 1000}, and iPrec@{0.1, 0.3, 0.7}R from the highest performing model configurations compared across the two query sets. The test statistic considers the individual differences of all the topics used in the experiments (not the averages). Results will be presented in table format.

Student's t-test. In order to substantiate the results from the Wilcoxon test, we carry out a further assessment of the differences using the Student's t-test. The t-test statistic is generally considered to be stronger than the non-parametric tests, but the assumptions that the data is sampled from a normal distribution and that it is measured on an interval scale have not been completely justified for the retrieval case Croft et al. (2009); Jones et al. (2000). Nonetheless, if an effectiveness measure satisfies the conditions for using the t-test, then the results will have more power than those of the non-parametric tests. The null hypothesis for a Student's t-test is that the *mean* difference between the pairs of observations is zero. We compute the t-test statistic and extract the appropriate P-values for the same measures selected for the Wilcoxon test, considering all the queries in the experiments. Results from the highest performing model configurations compared across the two query sets will be presented in table format.

Appendix B

Evaluation Dataset – Three Government Catalogues

Figures B.1-B.3 show the schemata extracted for the three catalogues of Public Sector Information (UK, US, and Australia). The three ontologies and respective instance data were combined into a single dataset for evaluation. The complete dataset can be downloaded from `http://catalogues.psi.enakting.org/psi/semsearch-eval` (14 Feb, 2013) as a single file in n-triples format.

An example record from the US catalogue is provided next. There are two sets of URI namespaces used in the data: one for global property and class definitions (e.g. `http://catalogues.psi.enakting.org/global/def/property/id/{name of property}`) and a local namespace for each catalogue used for instance data and local class definitions (e.g. a US dataset instance: `http://catalogues.psi.enakting.org/data.gov/dataset/id/1971`).

US catalogue data fragment, depicting a dataset record:

```
@prefix dc: <http://purl.org/dc/terms/> .
@prefix dcmi: <http://purl.org/dc/dcmitype/> .
@prefix medp: <http://catalogues.psi.enakting.org/global/def/property/id/> .
@prefix medc: <http://catalogues.psi.enakting.org/global/def/class/id/> .
@prefix usc: <http://catalogues.psi.enakting.org/data.gov/def/class/id/> .
@prefix agency: <http://catalogues.psi.enakting.org/data.gov/agency/id/> .
@prefix tag: <http://catalogues.psi.enakting.org/data.gov/tag/id/> .
@prefix cat: <http://catalogues.psi.enakting.org/data.gov/catalog/id/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .

### PSI Dataset Record Instance: ###

<http://catalogues.psi.enakting.org/data.gov/dataset/id/1971> a medc:Dataset;
    medp:partOf cat:main;
    dc:source <http://data.gov/details/1971>;
    dc:publisher agency:a1532584484;
    medp:tag tag:tag_bankruptcy;
    medp:tag tag:tag_counsel;
    medp:tag tag:tag_creditors;
    medp:tag tag:tag_debtors;
    medp:download-url <http://www.data.gov/download/1971/xls>;
    medp:category "Income, Expenditures, Poverty, and Wealth";
    medp:date-released "2010-01-01";
    medp:date-updated "10-Mar-2010";
    medp:time-period "Fiscal year";
    medp:update-frequency "Annual";
    dc:description "The annual data tables contained in this document provide
        summary statistics on the civil enforcement activities of the United
        States Trustee Program. These tables summarize data for Fiscal Year 2004
         through Fiscal Year 2009 (October 1, 2003 through September 30, 2009).
        This report covers informal actions. Informal actions include documented
         telephone calls, letters, e-mail transmissions, facsimile transmissions
        , and documented personal conversations by the United States Trustee to
        the debtor, debtor's counsel, or a third party (such as a trustee),
        raising a specific issue in a specific case that requires a response.";
    medp:category-type "Raw Data Catalog";
    medp:specialized-data-category-designation "Research";
    medp:unit-of-analysis "Motions filed";
    dc:coverage "USTP Offices";
    medp:collection-mode "Person/Computer";
    medp:technical-documentation-url <http://www.justice.gov/ust/eo/
        public_affairs/data_files/enf_act/docs/
        USTP_Enforcement_Activity_Data_Codebook.pdf>;
    rdfs:label "Informal Enforcement Actions Fiscal Year 2009" .

### Agency, Tag, and Catalogue Instances: ###

agency:a1532584484 a usc:Agency;
    rdfs:label "Department of Justice" .
```

```
tag:tag_bankruptcy a medc:Tag;
    rdfs:label "bankruptcy" .

tag:tag_counsel a medc:Tag;
    rdfs:label "counsel" .

tag:tag_creditors a medc:Tag;
    rdfs:label "creditors" .

tag:tag_debtors a medc:Tag;
    rdfs:label "debtors" .

cat:main a medc:Catalog;
    dc:source <http://data.gov/catalog>;
    rdfs:label "Data.gov -> US Government Raw and Tool Data Catalogue";
    dc:title "Data.gov -> US Government Raw and Tool Data Catalogue";
    dc:description "The raw and tool data catalogues of the United States'
        nationwide portal to PSI.";
    dc:hasPart <http://catalogues.psi.enakting.org/data.gov/dataset/id/1971>;
    ... .

### Schema Definitions: ###

medc:Dataset rdfs:label "Dataset";
    rdfs:subClassOf dcmi:Dataset .

medc:Catalog rdfs:label "Catalog";
    rdfs:subClassOf dcmi:Collection .

medc:Tag rdfs:label "Tag";
    rdf:type rdfs:Class .

usc:Agency rdfs:label "US Agency";
    rdfs:subClassOf dc:Agent .

medp:tag rdfs:label "Tag";
    rdf:type owl:ObjectProperty .

medp:partOf rdfs:label "Part of";
    rdf:type owl:ObjectProperty .

medp:download-url rdfs:label "Download URL";
    rdf:type owl:ObjectProperty .

medp:date-released rdfs:label "Date released";
    rdf:type owl:DatatypeProperty .

medp:date-updated rdfs:label "Date updated";
    rdf:type owl:DatatypeProperty .

medp:time-period rdfs:label "Time Period";
```

```
    rdf:type owl:DatatypeProperty .

medp:category-type rdfs:label "Category Type";
    rdf:type owl:DatatypeProperty .

medp:collection-mode rdfs:label "Collection Mode";
    rdf:type owl:DatatypeProperty .

medp:update-frequency rdfs:label "Update frequency";
    rdf:type owl:DatatypeProperty .

medp:specialized-data-category-designation rdfs:label "Specialised data
    category designation";
    rdf:type owl:DatatypeProperty .

medp:unit-of-analysis rdfs:label "Unit of analysis";
    rdf:type owl:DatatypeProperty .

medp:technical-documentation rdfs:label "Technical documentation description";
    rdf:type owl:DatatypeProperty .
```

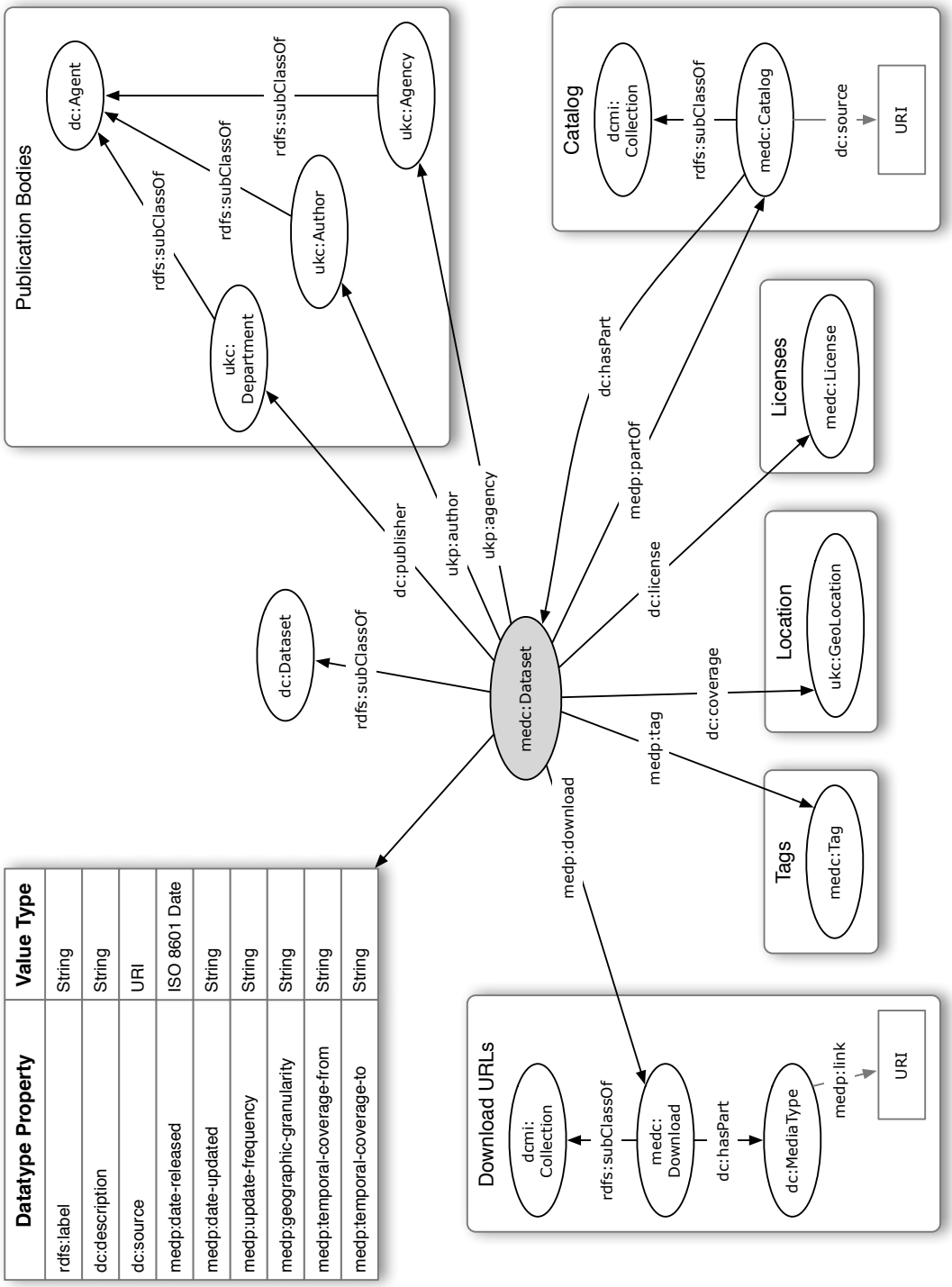


Figure B.1: A lightweight schema for the UK catalogue.

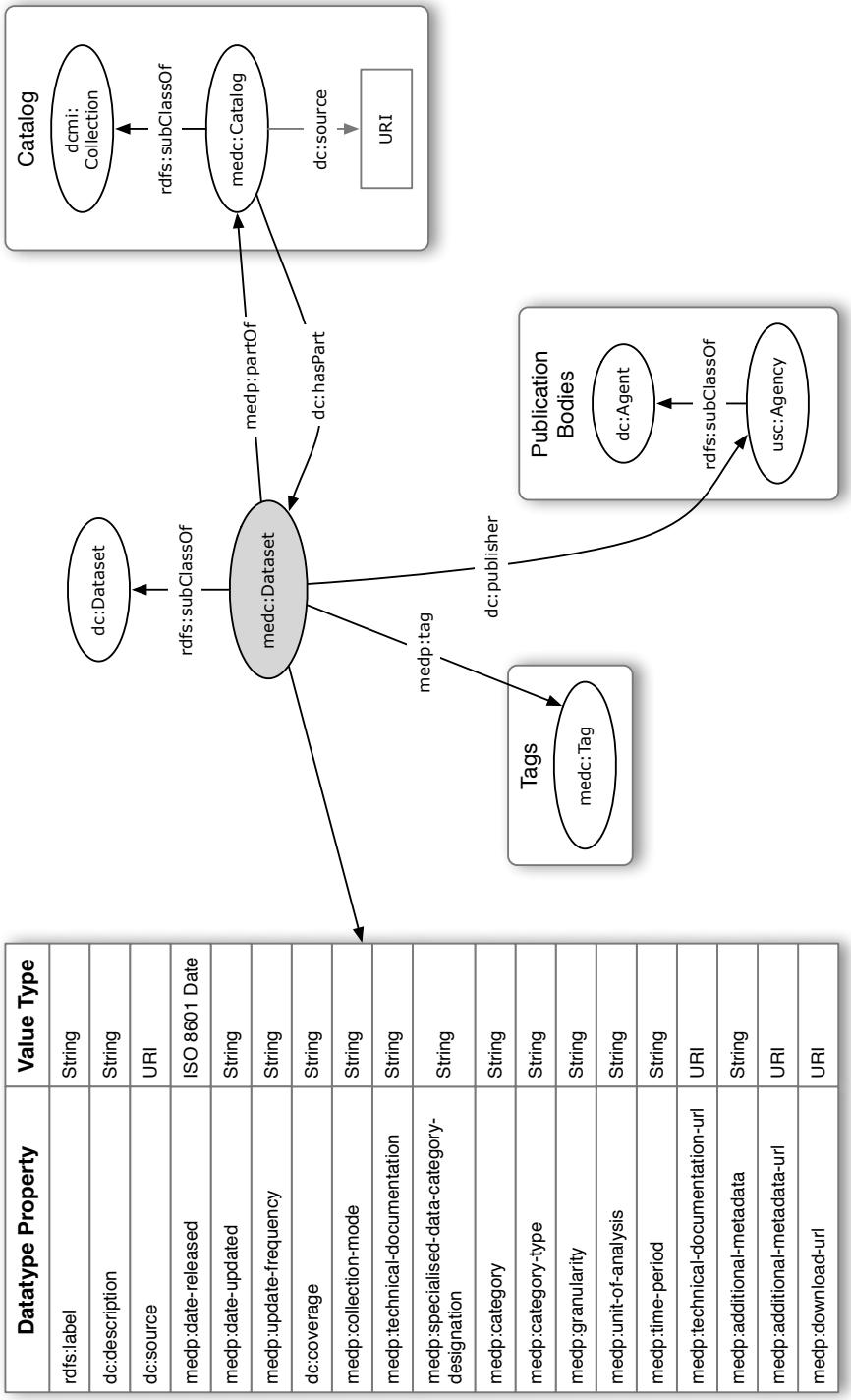


Figure B.2: A lightweight schema for the US catalogue.

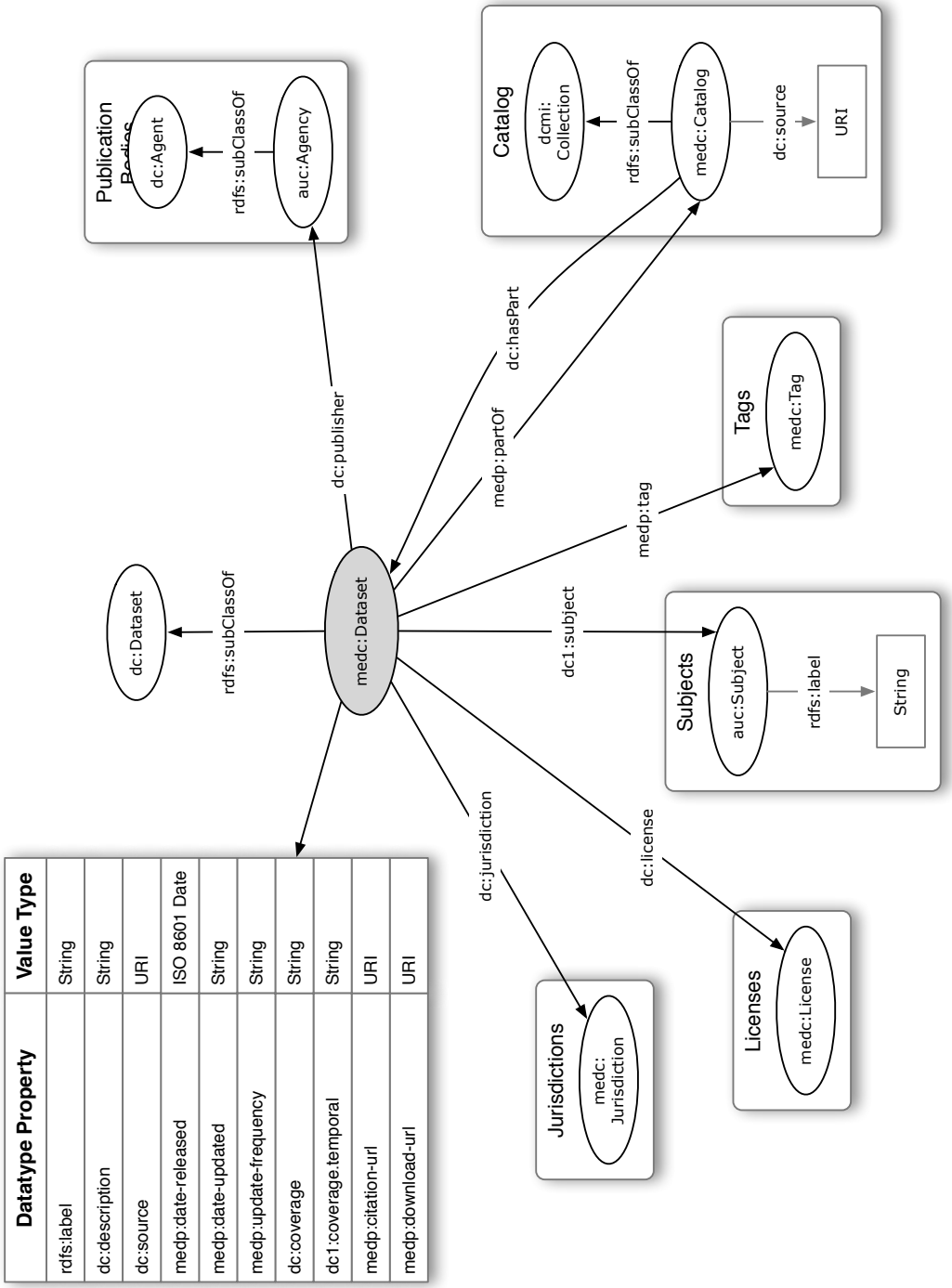


Figure B.3: A lightweight schema for the Australian catalogue.

Appendix C

Evaluation Topics and Per-Query Results

C.1 50 Topics for Assessment

Following are the 50 topics crafted for evaluation of the model on the catalogues dataset. The list of topics and the prepared "qrels" file needed for the TREC evaluation toolkit (Section 6.3.4) are available for download at `http://catalogues.psi.enakting.org/psi/semsearch-eval` (14 Feb, 2013).

Topic 1:

```
{
    "num": 1,
    "description": "Find all datasets released as part of data.gov.uk",
    "q1-simple": "data.gov.uk datasets",
    "q2-expressive": "datasets part of data.gov.uk",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> }"
}
```

Topic 2:

```
{
```

```
    "num": 2,
    "description": "Find all datasets released under the ukcrown license. This
        includes datasets with and without copyrights.",
    "q1-simple": "ukcrown datasets",
    "q2-expressive": "datasets licensed with ukcrown",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { { ?x <http://purl.org/dc/terms/
        license> <http://catalogues.psi.enakting.org/data.gov.uk/license/id/
        license-377997343> } UNION { ?x <http://purl.org/dc/terms/license> <http
        ://catalogues.psi.enakting.org/data.gov.uk/license/id/license-877972599>
         } }"
}
```

Topic 3:

```
{
    "num": 3,
    "description": "Find all datasets released as part of data.gov.uk and
        updated on a weekly basis.",
    "q1-simple": "data.gov.uk datasets updated weekly",
    "q2-expressive": "datasets part of data.gov.uk updated weekly",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/update-frequency> "Weekly" }"
}
```

Topic 4:

```
{
    "num": 4,
    "description": "Find all datasets released as part of data.gov.uk and are
        subject to daily updates.",
    "q1-simple": "data.gov.uk datasets updated daily",
    "q2-expressive": "datasets part of data.gov.uk updated daily",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/update-frequency> "Daily" }"
}
```

Topic 5:

```
{
    "num": 5,
    "description": "Find all datasets released as part of data.gov.uk during
        2008.",
    "q1-simple": "2008 data.gov.uk datasets",
    "q2-expressive": "datasets part of data.gov.uk released in 2008",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2008.*$", "
        i") }"
```

```
}
```

Topic 6:

```
{
    "num": 6,
    "description": "Find all datasets released as part of data.gov.uk catalog
        during 2009.",
    "q1-simple": "2009 data.gov.uk datasets",
    "q2-expressive": "datasets part of data.gov.uk released in 2009",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2009.*$", "
        i") }"
}
```

Topic 7:

```
{
    "num": 7,
    "description": "Find all datasets released as part of data.gov.uk during
        2010.",
    "q1-simple": "2010 data.gov.uk datasets",
    "q2-expressive": "datasets part of data.gov.uk released in 2010",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "
        i") }"
}
```

Topic 8:

```
{
    "num": 8,
    "description": "Find all departments referenced in data.gov.uk.",
    "q1-simple": "departments in data.gov.uk",
    "q2-expressive": "departments in data.gov.uk",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?y <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.uk/catalog/id/main> . ?y <http://purl.org/dc/terms/publisher> ?
        x . ?x a <http://catalogues.psi.enakting.org/data.gov.uk/def/class/id/
        Department> }"
}
```

Topic 9:

```
{
    "num": 9,
    "description": "Find all datasets published by the Welsh Assembly Government
        .",
```

```
    "q1-simple": "Welsh Assembly Government datasets",
    "q2-expressive": "datasets published by the Welsh Assembly Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-684976386> }"
}
```


Topic 10:


```
{
    "num": 10,
    "description": "Find all datasets authored by the Welsh Assembly Government
        .",
    "q1-simple": "Welsh Assembly Government datasets",
    "q2-expressive": "datasets authored by the Welsh Assembly Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/data.gov.uk/def/property/id/author> <http://catalogues.psi.enakting.
        org/data.gov.uk/author/id/hc-684976386> }"
}
```


Topic 11:


```
{
    "num": 11,
    "description": "Find all datasets published by the Northern Ireland
        Executive branch of government.",
    "q1-simple": "Northern Ireland Executive datasets",
    "q2-expressive": "datasets published by Northern Ireland Executive",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc250147435> }"
}
```


Topic 12:


```
{
    "num": 12,
    "description": "Find all datasets published by the Scottish Government.",
    "q1-simple": "Scottish Government datasets",
    "q2-expressive": "datasets published by Scottish Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-245904526> }"
}
```


Topic 13:


```
{
    "num": 13,
    "description": "Find all datasets published by the Scottish Government and
        released in 2010.",
    "q1-simple": "Scottish Government 2010 datasets",
```

```
    "q2-expressive": "datasets released in 2010 and published by Scottish
        Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-245904526> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "i") }"
}
```


Topic 14:


```
{
    "num": 14,
    "description": "Find all datasets published by DirectGov.",
    "q1-simple": "DirectGov datasets",
    "q2-expressive": "datasets published by DirectGov",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-962592763> }"
}
```


Topic 15:


```
{
    "num": 15,
    "description": "Find all datasets authored by DCLG Floor Targets Interactive
        .",
    "q1-simple": "DCLG Floor Targets Interactive datasets",
    "q2-expressive": "datasets authored by DCLG Floor Targets Interactive",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/data.gov.uk/def/property/id/author> <http://catalogues.psi.enakting.
        org/data.gov.uk/author/id/hc-622337974> }"
}
```


Topic 16:


```
{
    "num": 16,
    "description": "Find all datasets authored by DCLG Floor Targets Interactive
         and released during 2009.",
    "q1-simple": "DCLG Floor Targets Interactive 2009 datasets",
    "q2-expressive": "datasets released in 2009 and authored by DCLG Floor
        Targets Interactive",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/data.gov.uk/def/property/id/author> <http://catalogues.psi.enakting.
        org/data.gov.uk/author/id/hc-622337974> . ?x <http://catalogues.psi.
        enakting.org/global/def/property/id/date-released> ?o . FILTER regex(?o,
         "2009.*$", "i") }"
}
```


Topic 17:


```
{
```

```
    "num": 17,
    "description": "Find all datasets published by UK's Department for Work and
        Pensions.",
    "q1-simple": "UK Department for Work and Pensions datasets",
    "q2-expressive": "datasets published by UK Department for Work and Pensions
        ",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1449353688> }"
}
```


Topic 18:


```
{
    "num": 18,
    "description": "Find all datasets published by UK's Department for Work and
        Pensions in 2009.",
    "q1-simple": "UK Department for Work and Pensions 2009 datasets",
    "q2-expressive": "datasets released in 2009 and published by UK Department
        for Work and Pensions",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1449353688> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2009.*$", "i") }"
}
```


Topic 19:


```
{
    "num": 19,
    "description": "Find all datasets published by UK's Department for Work and
        Pensions in 2010.",
    "q1-simple": "UK Department for Work and Pensions 2010 datasets",
    "q2-expressive": "datasets released in 2010 and published by UK Department
        for Work and Pensions",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1449353688> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "i") }"
}
```


Topic 20:


```
{
    "num": 20,
    "description": "Find all datasets published by the Sunderland City Council
        .",
    "q1-simple": "Sunderland City Council datasets",
    "q2-expressive": "datasets published by Sunderland City Council",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-1893907108> }"
}
```

Topic 21:

```
{
    "num": 21,
    "description": "Find all datasets published by the Sunderland City Council
        in 2008.",
    "q1-simple": "UK Sunderland City Council 2008 datasets",
    "q2-expressive": "datasets released in 2008 and published by UK Sunderland
        City Council",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc-1893907108> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2008.*$", "i") }"
}
```

Topic 22:

```
{
    "num": 22,
    "description": "Find all datasets published by UK's Department for
        Environment, Food and Rural Affairs.",
    "q1-simple": "UK Department for Environment, Food and Rural Affairs datasets
        ",
    "q2-expressive": "datasets published by UK Department for Environment, Food
        and Rural Affairs",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc268762299> }"
}
```

Topic 23:

```
{
    "num": 23,
    "description": "Find all datasets published by UK's Department for
        Environment, Food and Rural Affairs during 2008.",
    "q1-simple": "UK Department for Environment, Food and Rural Affairs 2008
        datasets",
    "q2-expressive": "datasets released in 2008 and published by UK Department
        for Environment, Food and Rural Affairs",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc268762299> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2008.*$", "i") }"
}
```

Topic 24:

```
{
    "num": 24,
    "description": "Find all datasets published by UK's Department for
        Environment, Food and Rural Affairs during 2009.",
```

```
    "q1-simple": "UK Department for Environment, Food and Rural Affairs 2009
        datasets",
    "q2-expressive": "datasets released in 2009 and published by UK Department
        for Environment, Food and Rural Affairs",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc268762299> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2009.*$", "i") }"
}
```

Topic 25:

```
{
    "num": 25,
    "description": "Find all datasets published by UK's Department for
        Communities and Local Government.",
    "q1-simple": "UK Department for Communities and Local Government",
    "q2-expressive": "datasets published by UK Department for Communities and
        Local Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1960153517> }"
}
```

Topic 26:

```
{
    "num": 26,
    "description": "Find all datasets published by UK's Department for
        Communities and Local Government and are subject to Annual updates.",
    "q1-simple": "UK Department for Communities and Local Government annual
        datasets",
    "q2-expressive": "datasets updated annually and published by UK Department
        for Communities and Local Government",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1960153517> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/update-frequency> ?o . FILTER regex(?o, "annual.*$", "i") }"
}
```

Topic 27:

```
{
    "num": 27,
    "description": "Find all datasets published by Bristol City Council.",
    "q1-simple": "Bristol City Council datasets",
    "q2-expressive": "datasets published by Bristol City Council",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc1786920149> }"
}
```

Topic 28:

```
{
    "num": 28,
    "description": "Find all datasets published by Her Majesty's Revenue and
        Customs.",
    "q1-simple": "Her Majesty's Revenue and Customs datasets",
    "q2-expressive": "datasets published by Her Majesty's Revenue and Customs",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc247473247> }"
}
```

Topic 29:

```
{
    "num": 29,
    "description": "Find all datasets published by Her Majesty's Revenue and
        Customs and released in 2010.",
    "q1-simple": "Her Majesty's Revenue and Customs 2010 datasets",
    "q2-expressive": "datasets published by Her Majesty's Revenue and Customs
        and released in 2010.",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.uk/department/id
        /hc247473247> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "i") }"
}
```

Topic 30:

```
{
    "num": 30,
    "description": "Find all datasets released as part of the US catalog",
    "q1-simple": "US catalog datasets",
    "q2-expressive": "datasets part of US catalog",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> }"
}
```

Topic 31:

```
{
    "num": 31,
    "description": "Find all datasets released as part of data.gov during
        2008.",
    "q1-simple": "2008 data.gov datasets",
    "q2-expressive": "datasets part of data.gov released in 2008",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2008.*$", "
        i") }"
```

```
}
```

Topic 32:

```
{
    "num": 32,
    "description": "Find all datasets released as part of the US catalog during
        2009.",
    "q1-simple": "2009 US catalog datasets",
    "q2-expressive": "datasets part of US catalog released in 2009",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2009.*$", "
        i") }"
}
```

Topic 33:

```
{
    "num": 33,
    "description": "Find all datasets released as part of the US catalog during
        2010.",
    "q1-simple": "2010 US catalog datasets",
    "q2-expressive": "datasets part of US catalog released in 2010",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "
        i") }"
}
```

Topic 34:

```
{
    "num": 34,
    "description": "Find all datasets released as part of data.gov and updated
        continuously.",
    "q1-simple": "data.gov datasets updated continuously",
    "q2-expressive": "datasets part of data.gov updated continuously",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/update-frequency> ?o . FILTER regex(?o, "
        continuous.*$", "i") }"
}
```

Topic 35:

```
{
    "num": 35,
    "description": "Find all datasets released as part of the US catalog and
        updated annually.",
```

```
    "q1-simple": "US catalog datasets updated annually",
    "q2-expressive": "datasets part of US catalog updated annually",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/update-frequency> ?o . FILTER regex(?o, "annual.*
        $", "i") }"
}
```


Topic 36:


```
{
    "num": 36,
    "description": "Find all datasets released as part of data.gov and
        designated as administrative material.",
    "q1-simple": "data.gov Administrative",
    "q2-expressive": "datasets part of data.gov in administrative category",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov/catalog/id/main> . ?x <http://catalogues.psi.enakting.org/
        global/def/property/id/specialized-data-category-designation> ?o .
        FILTER regex(?o, "administrative.*$", "i") }"
}
```


Topic 37:


```
{
    "num": 37,
    "description": "Find all datasets published by the US Department of Defense
        .",
    "q1-simple": "US Department of Defense datasets",
    "q2-expressive": "datasets published by US Department of Defense",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/
        a32015941> }"
}
```


Topic 38:


```
{
    "num": 38,
    "description": "Find all datasets published by the US Department of Veterans
         Affairs.",
    "q1-simple": "US Department of Veterans Affairs datasets",
    "q2-expressive": "datasets published by US Department of Veterans Affairs",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/
        a418723769> }"
}
```


Topic 39:

```
{
    "num": 39,
    "description": "Find all datasets published by the Export-Import Bank of the
         US.",
    "q1-simple": "Export-Import Bank of the US datasets",
    "q2-expressive": "datasets published by Export-Import Bank of the US",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -243719468> }"
}
```


Topic 40:


```
{
    "num": 40,
    "description": "Find all datasets published by the US Institute of Museum
        and Library Services.",
    "q1-simple": "US Institute of Museum and Library Services datasets",
    "q2-expressive": "datasets published by US Institute of Museum and Library
        Services",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -66804868> }"
}
```


Topic 41:


```
{
    "num": 41,
    "description": "Find all datasets published by the US Equal Employment
        Opportunity Commission during 2010.",
    "q1-simple": "US Equal Employment Opportunity Commission 2010 datasets",
    "q2-expressive": "datasets released in 2010 and published by US Equal
        Employment Opportunity Commission",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -1728756322> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2010.*$", "i") }"
}
```


Topic 42:


```
{
    "num": 42,
    "description": "Find all datasets published by the US National
        Transportation Safety Board.",
    "q1-simple": "US National Transportation Safety Board datasets",
    "q2-expressive": "datasets published by US National Transportation Safety
        Board",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -303687458> }"
}
```

Topic 43:


```
{
    "num": 43,
    "description": "Find all administrative datasets published by the US
        National Transportation Safety Board.",
    "q1-simple": "US National Transportation Safety Board administrative
        datasets",
    "q2-expressive": "datasets published by US National Transportation Safety
        Board in category administrative",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -303687458> . ?x <http://catalogues.psi.enakting.org/global/def/property
        /id/specialized-data-category-designation> ?o . FILTER regex(?o, "
        administrative.*$", "i") }"
}
```


Topic 44:


```
{
    "num": 44,
    "description": "Find all datasets published by the US Environmental
        Protection Agency.",
    "q1-simple": "US Environmental Protection Agency datasets",
    "q2-expressive": "datasets published by US Environmental Protection Agency",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -1093567254> }"
}
```


Topic 45:


```
{
    "num": 45,
    "description": "Find all datasets published by the US Environmental
        Protection Agency during 2008.",
    "q1-simple": "US Environmental Protection Agency 2008 datasets",
    "q2-expressive": "datasets released in 2008 and published by US
        Environmental Protection Agency",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov/agency/id/a
        -1093567254> . ?x <http://catalogues.psi.enakting.org/global/def/
        property/id/date-released> ?o . FILTER regex(?o, "2008.*$", "i") }"
}
```


Topic 46:


```
{
    "num": 46,
    "description": "Find all datasets released as part of data.australia.gov.au
        ",
    "q1-simple": "data.australia.gov.au datasets",
    "q2-expressive": "datasets part of data.australia.gov.au",
```

```
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://catalogues.psi.enakting.
        org/global/def/property/id/partOf> <http://catalogues.psi.enakting.org/
        data.gov.au/catalog/id/main> }"
}
```

Topic 47:

```
{
    "num": 47,
    "description": "Find all datasets published by the Australian Territory and
        Municipal Services.",
    "q1-simple": "Australian Territory and Municipal Services datasets",
    "q2-expressive": "datasets published by Australian Territory and Municipal
        Services",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.au/agency/id/
        a1421660274> }"
}
```

Topic 48:

```
{
    "num": 48,
    "description": "Find all datasets published by the Australian Sustainability
         Victoria.",
    "q1-simple": "Australian Sustainability Victoria datasets",
    "q2-expressive": "datasets published by Australian Sustainability Victoria",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.au/agency/id/a
        -1479037704> }"
}
```

Topic 49:

```
{
    "num": 49,
    "description": "Find all datasets published by the Australian Department of
        the Environment, Water, Heritage and the Arts.",
    "q1-simple": "Australian Department of the Environment, Water, Heritage and
        the Arts datasets",
    "q2-expressive": "datasets published by the Australian Department of the
        Environment, Water, Heritage and the Arts",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.au/agency/id/a
        -1084736677> }"
}
```

Topic 50:

```
{
    "num": 50,
    "description": "Find all datasets published by the Australian Territory and
        Municipal Services",
```

```
    "q1-simple": "Australian Territory and Municipal Services datasets",
    "q2-expressive": "datasets published by the Australian Territory and
        Municipal Services",
    "q3-sparql": "SELECT DISTINCT ?x WHERE { ?x <http://purl.org/dc/terms/
        publisher> <http://catalogues.psi.enakting.org/data.gov.au/agency/id/
        a1421660274> }"
}
```

C.2 Per-Query Evaluation Results

The following tables show individual topic evaluation results utilising the best configurations for the two query types (`q1-simple`, `q2-expressive`), as discussed in Chapter 6. Specifically, results are provided for configurations *C1* and *C2*, for the two query sets, respectively (see Sections 6.4.2/6.5 for details). The raw "results" files prepared for the TREC evaluation toolkit (Section 6.3.4) can be downloaded from `http://catalogues.psi.enakting.org/psi/semsearch-eval` (14 Feb, 2013).

	Topic 1		Topic 2		Topic 3		Topic 4		Topic 5	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	7566	7597	3937	6037	7947	7977	7929	7960	10291	10473
Number Relevant	4149	4149	3935	3935	7	7	23	23	328	328
Number Relevant & Retrieved	4149	4149	3935	3935	7	7	23	23	328	328
Mean Average Precision	1.0000	1.0000	1.0000	1.0000	0.8736	0.4371	1.0000	0.3799	0.1539	1.0000
R Precision	1.0000	1.0000	1.0000	1.0000	0.8571	0.2857	1.0000	0.2609	0.1311	1.0000
Reciprocal Rank	1.0000	1.0000	1.0000	1.0000	1.0000	0.3333	1.0000	0.0556	0.0217	1.0000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	0.5385	1.0000	0.5750	0.2458	1.0000
Interpolated Precision at 0.10 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.20 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.30 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.40 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.50 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.60 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.70 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.80 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 0.90 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Interpolated Precision at 1.00 Recall	1.0000	1.0000	1.0000	1.0000	0.8750	0.5385	1.0000	0.5750	0.2321	1.0000
Precision after 5 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.8000	0.4000	1.0000	0.0000	0.0000	1.0000
Precision after 10 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.7000	0.4000	1.0000	0.0000	0.0000	1.0000
Precision after 15 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.4667	0.4667	1.0000	0.0000	0.0000	1.0000
Precision after 20 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.3500	0.3500	1.0000	0.1500	0.0000	1.0000
Precision after 30 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.2333	0.2333	0.7667	0.4333	0.0000	1.0000
Precision after 100 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.0700	0.0700	0.2300	0.2300	0.1400	1.0000
Precision after 200 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.0350	0.0350	0.1150	0.1150	0.1800	1.0000
Precision after 500 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.0140	0.0140	0.0460	0.0460	0.1160	0.6560
Precision after 1000 Docs Retrieved	1.0000	1.0000	1.0000	1.0000	0.0070	0.0070	0.0230	0.0230	0.1050	0.3280

Table C.2: Topics 1-5 evaluation results for configurations *C1* and *C2*.

	Topic 6		Topic 7		Topic 8		Topic 9		Topic 10	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	10337	10547	9203	9443	7617	7617	6071	6076	6071	6391
Number Relevant	2187	2187	1323	1323	67	67	212	212	213	213
Number Relevant & Retrieved	2187	2187	1323	1323	13	13	212	212	213	213
Mean Average Precision	0.7646	0.9968	0.5915	0.9982	0.0002	0.0002	0.9975	0.9894	0.9972	0.9909
R Precision	0.7577	0.9936	0.5072	0.9992	0.0000	0.0000	0.9953	0.9953	0.9953	0.9953
Reciprocal Rank	1.0000	1.0000	0.5000	1.0000	0.0002	0.0002	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	0.8750	1.0000	0.0021	0.0021	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.10 Recall	0.8219	1.0000	0.8108	0.9992	0.0021	0.0021	1.0000	0.9953	1.0000	1.0000
Interpolated Precision at 0.20 Recall	0.7956	1.0000	0.7938	0.9992	0.0000	0.0000	1.0000	0.9953	1.0000	0.9953
Interpolated Precision at 0.30 Recall	0.7956	1.0000	0.6616	0.9992	0.0000	0.0000	1.0000	0.9953	1.0000	0.9953
Interpolated Precision at 0.40 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	1.0000	0.9953	1.0000	0.9953
Interpolated Precision at 0.50 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	1.0000	0.9953	1.0000	0.9953
Interpolated Precision at 0.60 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	0.9953	0.9953	0.9953	0.9953
Interpolated Precision at 0.70 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	0.9953	0.9953	0.9953	0.9953
Interpolated Precision at 0.80 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	0.9953	0.9953	0.9953	0.9953
Interpolated Precision at 0.90 Recall	0.7956	0.9961	0.6616	0.9992	0.0000	0.0000	0.9953	0.9953	0.9953	0.9953
Interpolated Precision at 1.00 Recall	0.7956	0.9936	0.6605	0.9992	0.0000	0.0000	0.9953	0.9953	0.9383	0.9771
Precision after 5 Docs Retrieved	1.0000	1.0000	0.8000	1.0000	0.0000	0.0000	1.0000	1.0000	1.0000	1.0000
Precision after 10 Docs Retrieved	1.0000	1.0000	0.8000	1.0000	0.0000	0.0000	1.0000	1.0000	1.0000	1.0000
Precision after 15 Docs Retrieved	1.0000	1.0000	0.8000	1.0000	0.0000	0.0000	1.0000	1.0000	1.0000	1.0000
Precision after 20 Docs Retrieved	1.0000	1.0000	0.6000	1.0000	0.0000	0.0000	1.0000	1.0000	1.0000	1.0000
Precision after 30 Docs Retrieved	0.9333	1.0000	0.6667	1.0000	0.0000	0.0000	1.0000	0.9667	1.0000	1.0000
Precision after 100 Docs Retrieved	0.8900	1.0000	0.6900	1.0000	0.0000	0.0000	1.0000	0.9900	1.0000	0.9900
Precision after 200 Docs Retrieved	0.8050	1.0000	0.7900	0.9950	0.0000	0.0000	0.9950	0.9950	0.9950	0.9950
Precision after 500 Docs Retrieved	0.7600	1.0000	0.6180	0.9980	0.0000	0.0000	0.4240	0.4240	0.4260	0.4260
Precision after 1000 Docs Retrieved	0.7670	0.9940	0.4010	0.9990	0.0000	0.0000	0.2120	0.2120	0.2130	0.2130

Table C.4: Topics 6-10 evaluation results for configurations *C1* and *C2*.

	Topic 11		Topic 12		Topic 13		Topic 14		Topic 15	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	6087	6092	6105	6110	7887	8136	6052	6057	6047	6366
Number Relevant	130	130	264	264	125	125	5	5	226	226
Number Relevant & Retrieved	130	130	264	264	125	125	5	5	226	226
Mean Average Precision	0.9524	0.9554	0.7462	0.7065	0.6028	0.7576	0.0095	0.0020	0.9962	0.9905
R Precision	0.8615	0.8538	0.6402	0.6326	0.6960	0.7120	0.0000	0.0000	0.9956	0.9912
Reciprocal Rank	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0032	0.0007	1.0000	1.0000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.0158	0.0034	1.0000	1.0000
Interpolated Precision at 0.10 Recall	1.0000	1.0000	0.9773	0.9744	1.0000	1.0000	0.0158	0.0034	1.0000	1.0000
Interpolated Precision at 0.20 Recall	1.0000	1.0000	0.9412	0.9241	0.8387	1.0000	0.0158	0.0034	1.0000	0.9955
Interpolated Precision at 0.30 Recall	1.0000	1.0000	0.9345	0.9241	0.7311	0.8515	0.0158	0.0034	1.0000	0.9955
Interpolated Precision at 0.40 Recall	1.0000	1.0000	0.9345	0.9241	0.7311	0.8515	0.0158	0.0034	1.0000	0.9955
Interpolated Precision at 0.50 Recall	1.0000	1.0000	0.9345	0.9241	0.7311	0.8515	0.0158	0.0034	0.9956	0.9955
Interpolated Precision at 0.60 Recall	0.9898	1.0000	0.9191	0.9191	0.7311	0.8515	0.0158	0.0034	0.9956	0.9955
Interpolated Precision at 0.70 Recall	0.9898	0.9895	0.4835	0.3976	0.4944	0.8224	0.0158	0.0034	0.9956	0.9955
Interpolated Precision at 0.80 Recall	0.9455	0.9211	0.4835	0.3976	0.2665	0.4587	0.0158	0.0034	0.9956	0.9955
Interpolated Precision at 0.90 Recall	0.8681	0.8621	0.4835	0.3976	0.2665	0.3918	0.0158	0.0034	0.9956	0.9955
Interpolated Precision at 1.00 Recall	0.1443	0.1427	0.4835	0.3976	0.2665	0.3918	0.0158	0.0034	0.9956	0.9826
Precision after 5 Docs Retrieved	1.0000	1.0000	0.8000	0.8000	1.0000	1.0000	0.0000	0.0000	1.0000	1.0000
Precision after 10 Docs Retrieved	1.0000	1.0000	0.9000	0.9000	1.0000	1.0000	0.0000	0.0000	1.0000	1.0000
Precision after 15 Docs Retrieved	1.0000	1.0000	0.9333	0.9333	1.0000	1.0000	0.0000	0.0000	1.0000	1.0000
Precision after 20 Docs Retrieved	1.0000	1.0000	0.9500	0.9500	0.9500	1.0000	0.0000	0.0000	1.0000	1.0000
Precision after 30 Docs Retrieved	1.0000	1.0000	0.9667	0.9667	0.8333	0.9667	0.0000	0.0000	1.0000	0.9667
Precision after 100 Docs Retrieved	0.9700	0.9700	0.9000	0.8900	0.6800	0.8500	0.0000	0.0000	0.9900	0.9900
Precision after 200 Docs Retrieved	0.6350	0.6350	0.8150	0.8150	0.4450	0.4850	0.0000	0.0000	0.9950	0.9950
Precision after 500 Docs Retrieved	0.2540	0.2580	0.4360	0.3480	0.2500	0.2500	0.0100	0.0000	0.4520	0.4520
Precision after 1000 Docs Retrieved	0.1300	0.1300	0.2640	0.2640	0.1250	0.1250	0.0050	0.0000	0.2260	0.2260

Table C.6: Topics 11-15 evaluation results for configurations *C1* and *C2*.

	Topic 16		Topic 17		Topic 18		Topic 19		Topic 20	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	9071	9600	6620	6625	9627	9846	8364	8607	6151	6156
Number Relevant	226	226	125	125	89	89	30	30	42	42
Number Relevant & Retrieved	226	226	125	125	89	89	30	30	42	42
Mean Average Precision	0.9971	0.4852	0.3046	0.3307	0.4476	0.4776	0.1482	0.3342	0.4403	0.4466
R Precision	0.9956	0.2788	0.3520	0.3520	0.4382	0.5169	0.2000	0.3000	0.4286	0.4286
Reciprocal Rank	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.3333	1.0000	0.5000	0.5000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.3636	1.0000	0.5517	0.5667
Interpolated Precision at 0.10 Recall	1.0000	1.0000	0.5667	0.5600	0.9231	0.9286	0.3636	0.7500	0.5517	0.5667
Interpolated Precision at 0.20 Recall	1.0000	0.5597	0.3763	0.4464	0.8261	0.7500	0.2258	0.6000	0.5517	0.5667
Interpolated Precision at 0.30 Recall	1.0000	0.5597	0.3763	0.3643	0.6429	0.6585	0.1765	0.3600	0.5517	0.5667
Interpolated Precision at 0.40 Recall	1.0000	0.5597	0.3763	0.3571	0.5735	0.5902	0.1412	0.3171	0.5068	0.5667
Interpolated Precision at 0.50 Recall	1.0000	0.5597	0.3763	0.3516	0.4087	0.5169	0.1366	0.2778	0.5068	0.5068
Interpolated Precision at 0.60 Recall	0.9956	0.5597	0.2266	0.2695	0.3333	0.3506	0.1366	0.2571	0.5068	0.5068
Interpolated Precision at 0.70 Recall	0.9956	0.5597	0.0972	0.2058	0.2019	0.2333	0.1366	0.2471	0.5068	0.5068
Interpolated Precision at 0.80 Recall	0.9956	0.5597	0.0927	0.1621	0.0861	0.1423	0.0859	0.2162	0.5068	0.5068
Interpolated Precision at 0.90 Recall	0.9956	0.5597	0.0652	0.0638	0.0850	0.1067	0.0207	0.0571	0.5067	0.4419
Interpolated Precision at 1.00 Recall	0.9912	0.5368	0.0634	0.0590	0.0428	0.0604	0.0207	0.0548	0.3066	0.2745
Precision after 5 Docs Retrieved	1.0000	1.0000	0.8000	0.8000	1.0000	1.0000	0.2000	0.6000	0.4000	0.4000
Precision after 10 Docs Retrieved	1.0000	1.0000	0.8000	0.8000	0.9000	0.9000	0.3000	0.6000	0.2000	0.3000
Precision after 15 Docs Retrieved	1.0000	1.0000	0.6667	0.6667	0.8667	0.8667	0.2667	0.5333	0.2000	0.2000
Precision after 20 Docs Retrieved	1.0000	1.0000	0.5500	0.6000	0.8500	0.8000	0.2500	0.4000	0.4000	0.4000
Precision after 30 Docs Retrieved	1.0000	0.9667	0.5667	0.5333	0.7000	0.7000	0.2000	0.3000	0.5333	0.5667
Precision after 100 Docs Retrieved	1.0000	0.3200	0.3300	0.3300	0.3900	0.4900	0.1300	0.2100	0.3800	0.3900
Precision after 200 Docs Retrieved	0.9950	0.2150	0.3600	0.3300	0.2700	0.2800	0.1100	0.1200	0.2100	0.2100
Precision after 500 Docs Retrieved	0.4520	0.4520	0.1720	0.1860	0.1300	0.1420	0.0500	0.0520	0.0840	0.0840
Precision after 1000 Docs Retrieved	0.2260	0.2260	0.0930	0.1060	0.0830	0.0850	0.0260	0.0300	0.0420	0.0420

Table C.8: Topics 16-20 evaluation results for configurations *C1* and *C2*.

	Topic 21		Topic 22		Topic 23		Topic 24		Topic 25	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	9521	9716	6569	6574	9576	9771	9577	9797	6632	6654
Number Relevant	18	18	238	238	31	31	83	83	293	293
Number Relevant & Retrieved	18	18	238	238	31	31	83	83	293	293
Mean Average Precision	0.7094	0.9940	0.9591	0.9485	0.1965	0.9508	0.3823	0.6741	0.8078	0.7650
R Precision	0.6111	0.9444	0.8571	0.8529	0.2258	0.9355	0.3373	0.5904	0.6894	0.6621
Reciprocal Rank	1.0000	1.0000	1.0000	1.0000	0.3333	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	1.0000	1.0000	0.3333	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.10 Recall	1.0000	1.0000	1.0000	1.0000	0.3043	1.0000	0.5833	1.0000	0.9500	0.9412
Interpolated Precision at 0.20 Recall	0.8333	1.0000	1.0000	1.0000	0.3043	1.0000	0.5122	0.9500	0.9444	0.9263
Interpolated Precision at 0.30 Recall	0.8333	1.0000	1.0000	1.0000	0.2479	1.0000	0.4182	0.8929	0.9444	0.9263
Interpolated Precision at 0.40 Recall	0.8333	1.0000	1.0000	1.0000	0.2479	1.0000	0.4182	0.7727	0.9252	0.8966
Interpolated Precision at 0.50 Recall	0.8333	1.0000	1.0000	1.0000	0.2479	0.9667	0.4182	0.6125	0.9153	0.8902
Interpolated Precision at 0.60 Recall	0.6875	1.0000	1.0000	0.9892	0.2479	0.9667	0.3571	0.5862	0.8685	0.8558
Interpolated Precision at 0.70 Recall	0.6522	1.0000	1.0000	0.9892	0.2479	0.9667	0.3571	0.5447	0.6624	0.5695
Interpolated Precision at 0.80 Recall	0.6522	1.0000	0.9750	0.9697	0.2479	0.9667	0.3071	0.5447	0.5732	0.5096
Interpolated Precision at 0.90 Recall	0.4359	0.9474	0.7256	0.6618	0.2479	0.9667	0.2282	0.3028	0.5593	0.4719
Interpolated Precision at 1.00 Recall	0.2045	0.9474	0.7256	0.6611	0.1403	0.5254	0.1912	0.2065	0.5260	0.3644
Precision after 5 Docs Retrieved	0.8000	1.0000	1.0000	1.0000	0.2000	1.0000	0.6000	1.0000	1.0000	1.0000
Precision after 10 Docs Retrieved	0.8000	1.0000	1.0000	1.0000	0.1000	1.0000	0.4000	1.0000	1.0000	1.0000
Precision after 15 Docs Retrieved	0.6667	1.0000	1.0000	1.0000	0.2000	0.9333	0.4667	1.0000	1.0000	1.0000
Precision after 20 Docs Retrieved	0.6000	0.9000	1.0000	1.0000	0.3000	0.9500	0.5000	0.9500	0.9500	0.9500
Precision after 30 Docs Retrieved	0.5333	0.6000	1.0000	1.0000	0.2333	0.9667	0.5333	0.8667	0.9333	0.9333
Precision after 100 Docs Retrieved	0.1800	0.1800	1.0000	1.0000	0.1300	0.3100	0.3600	0.5400	0.9400	0.9000
Precision after 200 Docs Retrieved	0.0900	0.0900	0.9750	0.9600	0.1450	0.1550	0.3100	0.3700	0.8700	0.8600
Precision after 500 Docs Retrieved	0.0360	0.0360	0.4760	0.4760	0.0620	0.0620	0.1660	0.1660	0.5480	0.5000
Precision after 1000 Docs Retrieved	0.0180	0.0180	0.2380	0.2380	0.0310	0.0310	0.0830	0.0830	0.2930	0.2930

Table C.10: Topics 21-25 evaluation results for configurations *C1* and *C2*.

	Topic 26		Topic 27		Topic 28		Topic 29		Topic 30	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	6807	7166	6169	6174	6053	6058	7839	8089	6196	6233
Number Relevant	117	117	15	15	141	141	45	45	1779	1779
Number Relevant & Retrieved	117	117	15	15	141	141	45	45	1779	1779
Mean Average Precision	0.3101	0.4018	0.4969	0.4862	0.9422	0.9362	0.8979	0.9643	1.0000	1.0000
R Precision	0.3077	0.4017	0.4000	0.4000	0.9291	0.9291	0.8889	0.9111	1.0000	1.0000
Reciprocal Rank	1.0000	0.0667	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.00 Recall	1.0000	0.5294	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.10 Recall	0.4615	0.5294	1.0000	1.0000	0.9924	0.9922	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.20 Recall	0.4615	0.5294	1.0000	1.0000	0.9924	0.9922	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.30 Recall	0.3529	0.5294	1.0000	1.0000	0.9924	0.9922	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.40 Recall	0.2680	0.4335	1.0000	1.0000	0.9924	0.9922	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.50 Recall	0.2680	0.4335	0.2391	0.2105	0.9924	0.9922	0.9722	1.0000	1.0000	1.0000
Interpolated Precision at 0.60 Recall	0.2680	0.4335	0.2391	0.2000	0.9924	0.9922	0.9722	1.0000	1.0000	1.0000
Interpolated Precision at 0.70 Recall	0.2680	0.4335	0.2391	0.2000	0.9924	0.9922	0.9722	0.9756	1.0000	1.0000
Interpolated Precision at 0.80 Recall	0.2680	0.4317	0.1200	0.1071	0.9924	0.9922	0.9231	0.9756	1.0000	1.0000
Interpolated Precision at 0.90 Recall	0.2670	0.4280	0.0782	0.0718	0.9924	0.9922	0.3417	0.9111	1.0000	1.0000
Interpolated Precision at 1.00 Recall	0.2635	0.3128	0.0311	0.0314	0.3183	0.2981	0.0985	0.2432	1.0000	1.0000
Precision after 5 Docs Retrieved	0.4000	0.0000	1.0000	1.0000	0.8000	0.8000	1.0000	1.0000	1.0000	1.0000
Precision after 10 Docs Retrieved	0.5000	0.0000	0.6000	0.6000	0.9000	0.9000	1.0000	1.0000	1.0000	1.0000
Precision after 15 Docs Retrieved	0.4667	0.0667	0.4000	0.4000	0.9333	0.9333	1.0000	1.0000	1.0000	1.0000
Precision after 20 Docs Retrieved	0.4500	0.1000	0.3000	0.3000	0.9500	0.9500	1.0000	1.0000	1.0000	1.0000
Precision after 30 Docs Retrieved	0.4000	0.2333	0.2000	0.2000	0.9667	0.9667	0.9667	1.0000	1.0000	1.0000
Precision after 100 Docs Retrieved	0.3500	0.4500	0.1200	0.1100	0.9900	0.9900	0.4000	0.4400	1.0000	1.0000
Precision after 200 Docs Retrieved	0.2350	0.4250	0.0700	0.0700	0.6700	0.6700	0.2200	0.2250	1.0000	1.0000
Precision after 500 Docs Retrieved	0.2340	0.2340	0.0300	0.0300	0.2820	0.2820	0.0900	0.0900	1.0000	1.0000
Precision after 1000 Docs Retrieved	0.1170	0.1170	0.0150	0.0150	0.1410	0.1410	0.0450	0.0450	1.0000	1.0000

Table C.12: Topics 26-30 evaluation results for configurations *C1* and *C2*.

	Topic 31		Topic 32		Topic 33		Topic 34		Topic 35	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	9931	10120	9203	9459	7979	8259	7556	7588	6717	6754
Number Relevant	117	117	841	841	248	248	169	169	631	631
Number Relevant & Retrieved	117	117	841	841	248	248	169	169	631	631
Mean Average Precision	0.0129	0.5492	0.6029	0.8158	0.5633	0.9179	0.7110	0.8571	0.6065	0.6100
R Precision	0.0000	0.4615	0.5446	0.8121	0.6774	0.9234	0.7692	0.9349	0.7195	0.7385
Reciprocal Rank	0.0002	0.5000	1.0000	1.0000	0.3333	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.00 Recall	0.0230	0.9818	1.0000	1.0000	0.7377	1.0000	1.0000	1.0000	1.0000	1.0000
Interpolated Precision at 0.10 Recall	0.0230	0.9818	0.7389	1.0000	0.7377	0.9783	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.20 Recall	0.0230	0.9818	0.7389	1.0000	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.30 Recall	0.0230	0.9818	0.7389	0.8342	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.40 Recall	0.0230	0.9818	0.7389	0.8342	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.50 Recall	0.0230	0.2427	0.7342	0.8342	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.60 Recall	0.0230	0.2427	0.6300	0.8342	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.70 Recall	0.0230	0.2427	0.6300	0.8342	0.7377	0.9662	0.8649	0.9398	0.7720	0.7844
Interpolated Precision at 0.80 Recall	0.0230	0.2427	0.6300	0.8342	0.7377	0.9662	0.7949	0.9398	0.7720	0.7844
Interpolated Precision at 0.90 Recall	0.0230	0.2427	0.6300	0.8342	0.7377	0.9662	0.7949	0.9398	0.7720	0.7844
Interpolated Precision at 1.00 Recall	0.0230	0.1937	0.6300	0.8269	0.4429	0.4576	0.0387	0.5417	0.5981	0.2976
Precision after 5 Docs Retrieved	0.0000	0.8000	0.8000	1.0000	0.6000	1.0000	1.0000	0.8000	0.8000	1.0000
Precision after 10 Docs Retrieved	0.0000	0.9000	0.4000	1.0000	0.3000	1.0000	0.8000	0.8000	0.9000	0.9000
Precision after 15 Docs Retrieved	0.0000	0.9333	0.2667	1.0000	0.2000	1.0000	0.5333	0.5333	0.7333	0.7333
Precision after 20 Docs Retrieved	0.0000	0.9500	0.3000	1.0000	0.2000	0.9500	0.4000	0.5500	0.7000	0.7000
Precision after 30 Docs Retrieved	0.0000	0.9667	0.4000	1.0000	0.3667	0.9667	0.3333	0.6667	0.6333	0.4667
Precision after 100 Docs Retrieved	0.0000	0.5400	0.3700	1.0000	0.4100	0.9300	0.8000	0.9000	0.2400	0.1600
Precision after 200 Docs Retrieved	0.0000	0.2900	0.6050	0.9600	0.6000	0.9600	0.7750	0.8400	0.2750	0.2500
Precision after 500 Docs Retrieved	0.0000	0.2320	0.7180	0.7180	0.4520	0.4660	0.3180	0.3380	0.6460	0.6700
Precision after 1000 Docs Retrieved	0.0000	0.1170	0.5350	0.8270	0.2480	0.2480	0.1640	0.1690	0.6300	0.6280

Table C.14: Topics 31-35 evaluation results for configurations *C1* and *C2*.

	Topic 36		Topic 37		Topic 38		Topic 39		Topic 40	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	7208	7258	6275	6280	6287	6292	6250	6255	6481	6486
Number Relevant	965	965	204	204	27	27	4	4	18	18
Number Relevant & Retrieved	965	965	204	204	27	27	4	4	18	18
Mean Average Precision	0.1005	0.6763	0.6305	0.5475	0.2929	0.2918	0.5903	0.5903	0.7669	0.7669
R Precision	0.0000	0.6674	0.7304	0.6029	0.2222	0.2222	0.7500	0.7500	0.8889	0.8889
Reciprocal Rank	0.0002	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	0.5000	0.3333	0.3333
Interpolated Precision at 0.00 Recall	0.1871	1.0000	1.0000	1.0000	1.0000	1.0000	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.10 Recall	0.1871	0.9932	0.7809	0.7076	1.0000	1.0000	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.20 Recall	0.1871	0.9892	0.7809	0.7076	0.8571	0.8571	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.30 Recall	0.1871	0.9533	0.7809	0.7076	0.2667	0.2609	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.40 Recall	0.1871	0.9533	0.7809	0.7076	0.2667	0.2609	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.50 Recall	0.1871	0.9153	0.7809	0.7076	0.0769	0.0740	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.60 Recall	0.1871	0.8453	0.7809	0.7076	0.0769	0.0740	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.70 Recall	0.1871	0.1873	0.7809	0.7076	0.0769	0.0740	0.7500	0.7500	0.9000	0.9000
Interpolated Precision at 0.80 Recall	0.1871	0.1873	0.7809	0.7076	0.0769	0.0740	0.4444	0.4444	0.9000	0.9000
Interpolated Precision at 0.90 Recall	0.1871	0.1873	0.7809	0.7076	0.0504	0.0624	0.4444	0.4444	0.9000	0.9000
Interpolated Precision at 1.00 Recall	0.1871	0.1873	0.6711	0.6335	0.0504	0.0501	0.4444	0.4444	0.9000	0.9000
Precision after 5 Docs Retrieved	0.0000	0.8000	1.0000	1.0000	0.8000	0.8000	0.6000	0.6000	0.6000	0.6000
Precision after 10 Docs Retrieved	0.0000	0.9000	0.9000	0.8000	0.6000	0.6000	0.4000	0.4000	0.8000	0.8000
Precision after 15 Docs Retrieved	0.0000	0.9333	0.6000	0.5333	0.4000	0.4000	0.2667	0.2667	0.8667	0.8667
Precision after 20 Docs Retrieved	0.0000	0.9500	0.4500	0.4000	0.3000	0.3000	0.2000	0.2000	0.9000	0.9000
Precision after 30 Docs Retrieved	0.0000	0.9667	0.4333	0.3667	0.2000	0.2000	0.1333	0.1333	0.6000	0.6000
Precision after 100 Docs Retrieved	0.0000	0.9900	0.4500	0.2200	0.1200	0.1200	0.0400	0.0400	0.1800	0.1800
Precision after 200 Docs Retrieved	0.0000	0.9850	0.7250	0.5950	0.0600	0.0600	0.0200	0.0200	0.0900	0.0900
Precision after 500 Docs Retrieved	0.0000	0.9260	0.4080	0.4080	0.0480	0.0520	0.0080	0.0080	0.0360	0.0360
Precision after 1000 Docs Retrieved	0.0000	0.6440	0.2040	0.2040	0.0270	0.0270	0.0040	0.0040	0.0180	0.0180

Table C.16: Topics 36-40 evaluation results for configurations *C1* and *C2*.

	Topic 41		Topic 42		Topic 43		Topic 44		Topic 45	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	8068	8317	6457	6462	6480	6488	6376	6381	9399	9596
Number Relevant	12	12	22	22	7	7	464	464	67	67
Number Relevant & Retrieved	12	12	22	22	7	7	464	464	67	67
Mean Average Precision	0.1626	0.4215	0.5798	0.4999	0.3137	0.9087	0.9459	0.9219	0.7161	0.9958
R Precision	0.0833	0.0833	0.7273	0.6818	0.5714	0.7143	0.9591	0.9375	0.8507	0.9552
Reciprocal Rank	0.2500	1.0000	0.2500	0.2500	0.2500	1.0000	1.0000	1.0000	0.3333	1.0000
Interpolated Precision at 0.00 Recall	0.2500	1.0000	0.8000	0.6957	0.5714	1.0000	1.0000	1.0000	0.8906	1.0000
Interpolated Precision at 0.10 Recall	0.2449	0.5217	0.8000	0.6957	0.5714	1.0000	0.9922	0.9845	0.8906	1.0000
Interpolated Precision at 0.20 Recall	0.2449	0.5217	0.8000	0.6957	0.5714	1.0000	0.9922	0.9845	0.8906	1.0000
Interpolated Precision at 0.30 Recall	0.2449	0.5217	0.8000	0.6957	0.5714	1.0000	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.40 Recall	0.2449	0.5217	0.8000	0.6957	0.5714	1.0000	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.50 Recall	0.2449	0.5217	0.8000	0.6957	0.5714	1.0000	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.60 Recall	0.2449	0.5217	0.8000	0.6957	0.1724	0.8333	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.70 Recall	0.2449	0.5217	0.8000	0.6957	0.1724	0.8333	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.80 Recall	0.2449	0.5217	0.4130	0.3958	0.1556	0.7778	0.9607	0.9412	0.8906	1.0000
Interpolated Precision at 0.90 Recall	0.2449	0.5217	0.3860	0.3143	0.1556	0.7778	0.9607	0.9412	0.8356	0.9839
Interpolated Precision at 1.00 Recall	0.2449	0.5217	0.3860	0.3143	0.1556	0.7778	0.9607	0.9412	0.2114	0.9571
Precision after 5 Docs Retrieved	0.2000	0.2000	0.4000	0.4000	0.4000	0.8000	0.8000	0.8000	0.6000	1.0000
Precision after 10 Docs Retrieved	0.1000	0.1000	0.6000	0.4000	0.4000	0.7000	0.9000	0.8000	0.4000	1.0000
Precision after 15 Docs Retrieved	0.0667	0.2667	0.7333	0.5333	0.2667	0.4667	0.9333	0.8667	0.5333	1.0000
Precision after 20 Docs Retrieved	0.0500	0.4500	0.8000	0.6500	0.2000	0.3500	0.9500	0.9000	0.6500	1.0000
Precision after 30 Docs Retrieved	0.0333	0.4000	0.5667	0.5667	0.1667	0.2333	0.9667	0.9333	0.7667	1.0000
Precision after 100 Docs Retrieved	0.1200	0.1200	0.2200	0.2200	0.0700	0.0700	0.9900	0.9800	0.6100	0.6700
Precision after 200 Docs Retrieved	0.0600	0.0600	0.1100	0.1100	0.0350	0.0350	0.9050	0.8750	0.3250	0.3350
Precision after 500 Docs Retrieved	0.0240	0.0240	0.0440	0.0440	0.0140	0.0140	0.9280	0.9280	0.1340	0.1340
Precision after 1000 Docs Retrieved	0.0120	0.0120	0.0220	0.0220	0.0070	0.0070	0.4640	0.4640	0.0670	0.0670

Table C.18: Topics 41-45 evaluation results for configurations *C1* and *C2*.

	Topic 46		Topic 47		Topic 48		Topic 49		Topic 50	
	C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
Number Retrieved	7200	7232	6288	6293	6074	6079	6226	6231	6288	6293
Number Relevant	69	69	3	3	5	5	4	4	3	3
Number Relevant & Retrieved	69	69	3	3	5	5	4	4	3	3
Mean Average Precision	0.3417	0.3417	1.0000	1.0000	0.5229	0.5054	0.4659	0.5909	1.0000	1.0000
R Precision	0.3623	0.3623	1.0000	1.0000	0.4000	0.4000	0.5000	0.5000	1.0000	1.0000
Reciprocal Rank	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	1.0000
Interpolated Precision at 0.00 Recall	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	1.0000
Interpolated Precision at 0.10 Recall	0.9615	0.9615	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	1.0000
Interpolated Precision at 0.20 Recall	0.9615	0.9615	1.0000	1.0000	1.0000	1.0000	0.5000	1.0000	1.0000	1.0000
Interpolated Precision at 0.30 Recall	0.9615	0.9615	1.0000	1.0000	1.0000	1.0000	0.5000	0.5000	1.0000	1.0000
Interpolated Precision at 0.40 Recall	0.0115	0.0115	1.0000	1.0000	1.0000	1.0000	0.5000	0.5000	1.0000	1.0000
Interpolated Precision at 0.50 Recall	0.0115	0.0115	1.0000	1.0000	0.2857	0.2500	0.5000	0.5000	1.0000	1.0000
Interpolated Precision at 0.60 Recall	0.0115	0.0115	1.0000	1.0000	0.2857	0.2500	0.5000	0.5000	1.0000	1.0000
Interpolated Precision at 0.70 Recall	0.0115	0.0115	1.0000	1.0000	0.2857	0.2500	0.5000	0.5000	1.0000	1.0000
Interpolated Precision at 0.80 Recall	0.0115	0.0115	1.0000	1.0000	0.2857	0.2500	0.3636	0.3636	1.0000	1.0000
Interpolated Precision at 0.90 Recall	0.0115	0.0115	1.0000	1.0000	0.0980	0.0769	0.3636	0.3636	1.0000	1.0000
Interpolated Precision at 1.00 Recall	0.0115	0.0115	1.0000	1.0000	0.0980	0.0769	0.3636	0.3636	1.0000	1.0000
Precision after 5 Docs Retrieved	0.8000	0.8000	0.6000	0.6000	0.4000	0.4000	0.4000	0.4000	0.6000	0.6000
Precision after 10 Docs Retrieved	0.9000	0.9000	0.3000	0.3000	0.2000	0.2000	0.3000	0.3000	0.3000	0.3000
Precision after 15 Docs Retrieved	0.9333	0.9333	0.2000	0.2000	0.2667	0.2000	0.2667	0.2667	0.2000	0.2000
Precision after 20 Docs Retrieved	0.9500	0.9500	0.1500	0.1500	0.2000	0.2000	0.2000	0.2000	0.1500	0.1500
Precision after 30 Docs Retrieved	0.8333	0.8333	0.1000	0.1000	0.1333	0.1333	0.1333	0.1333	0.1000	0.1000
Precision after 100 Docs Retrieved	0.2500	0.2500	0.0300	0.0300	0.0500	0.0500	0.0400	0.0400	0.0300	0.0300
Precision after 200 Docs Retrieved	0.1250	0.1250	0.0150	0.0150	0.0250	0.0250	0.0200	0.0200	0.0150	0.0150
Precision after 500 Docs Retrieved	0.0500	0.0500	0.0060	0.0060	0.0100	0.0100	0.0080	0.0080	0.0060	0.0060
Precision after 1000 Docs Retrieved	0.0250	0.0250	0.0030	0.0030	0.0050	0.0050	0.0040	0.0040	0.0030	0.0030

Table C.20: Topics 46-50 evaluation results for configurations *C1* and *C2*.

Abbreviations

BN Bayesian Network

CPT Conditional Probability Table

CSV Comma Separated Value

DAG Directed Acyclic Graph

DLG Directed Labelled Graph

HTML HyperText Markup Language

IDF Inverse Collection/Document Frequency

INEX Initiative for the Evaluation of XML retrieval

iPrec Interpolated Precision - see Appendix A

IR Information Retrieval

LOD Linked Open Data

LS Literal Space of the Bayesian Network model

MAP Mean Average Precision - see Appendix A

MLE Maximum Likelihood Estimate

NDCG Non Discounted Cumulative Gain

P@10 Precision at 10 (after 10 documents have been retrieved) - see Appendix A

PSI Public Sector Information

RDBMS Relational Database Management System

RDF(a) Resource Description Framework (- in - attributes)

REGEX Regular Expression

SPARQL SPARQL Protocol and RDF Query Language

SW Semantic Web

TF Term Frequency

TREC Text Retrieval Evaluation Conference

URI Uniform Resource Identifier

XML Extensible Markup Language

Bibliography

Alani, H., Brewster, C., and Shadbolt, N. (2006). Ranking ontologies with aktiverank. In *The 5th International Semantic Web Conference (ISWC)*, volume 4273, pages 1–15. LNCS.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22(3):261–295.

Baeza-Yates, R. (2004). Challenges in the interaction of information retrieval and natural language processing. In *Computational Linguistics and Intelligent Text Processing*, volume 2945 of *Lecture Notes in Computer Science*, pages 445–456. Springer Berlin / Heidelberg. 10.1007/978-3-540-24630-5-55.

Baeza-Yates, R., Ciaramita, M., Mika, P., and Zaragoza, H. (2008). Towards semantic search. In *Proceedings of the 13th international conference on Natural Language and Information Systems: Applications of Natural Language to Information Systems*, NLDB '08, pages 4–11, Berlin, Heidelberg. Springer-Verlag.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Balmin, A., Hristidis, V., and Papakonstantinou, Y. (2004). Objectrank: authority-based keyword search in databases. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 564–575. VLDB Endowment.

Balog, K., Ciglan, M., Neumayer, R., Wei, W., and Nørvåg, K. (2011). Ntnu at semsearch 2011. In *4th International Semantic Search Workshop*, India.

Balog, K., Fang, Y., de Rijke, M., Serdyukov, P., and Si, L. (2012). Expertise retrieval. *Found. Trends Inf. Retr.*, 6(2-3):127–256.

Balog, K., Serdyukov, P., and Vries, A. (2010). Overview of the trec 2010 entity track. Technical report, DTIC Document.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific American*, 284(5):28–37.

Bhalotia, G., Hulgeri, A., Nakhe, C., Chakrabarti, S., and Sudarshan, S. (2002). Keyword searching and browsing in databases using banks. In *Proceedings of the 18th International Conference on Data Engineering*, pages 431–440. IEEE Computer Society.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 4. Springer New York.

Blanco, R., Mika, P., and Vigna, S. (2011). Effective and efficient entity search in rdf data. In *Proceedings of the 10th International Conference on the Semantic Web*, volume 7031 of *Lecture Notes in Computer Science*, pages 83–97. Springer Berlin / Heidelberg. 10.1007/978-3-642-25073-6-6.

Blanco, R., Mika, P., and Zaragoza, H. (2010). Entity search track submission by yahoo! research barcelona. In *3th International Semantic Search Workshop, USA*.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

Calado, P., Ribeiro-Neto, B., Ziviani, N., Moura, E., and Silva, I. (2003). Local versus global link information in the web. *ACM Trans. Inf. Syst.*, 21(1):42–63.

Celino, I., Valle, E. D., Cerizza, D., and Turati, A. (2006). Squiggle: a semantic search engine for indexing and retrieval of multimedia content. In *SEMPS*, volume 228 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Chen, H. and Lynch, K. (1992). Automatic construction of networks of concepts characterizing document databases. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(5):885 –902.

Chen, H. and Ng, T. (1995). An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist hopfield net activation. *Journal of the American Society for Information Science*, 46:348–369.

Chen, Y., Wang, W., Liu, Z., and Lin, X. (2009). Keyword search on structured and semi-structured data. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 1005–1010, New York, NY, USA. ACM.

Cohen, S., Mamou, J., Kanza, Y., and Sagiv, Y. (2003). Xsearch: a semantic search engine for xml. In *Proceedings of the 29th international conference on Very large data bases - Volume 29*, VLDB '03, pages 45–56. VLDB Endowment.

Cooper, G. F. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42(2-3):393 – 405.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482.

Crestani, F., de Campos, L. M., Fernández-Luna, J. M., and Huete, J. F. (2003). A multi-layered bayesian network model for structured document retrieval. In *ECSQARU'03*, pages 74–86.

Croft, B., Metzler, D., and Strohman, T. (2009). *Search Engines: Information Retrieval in Practice*. Addison Wesley, 1 edition.

Croft, B. and Turtle, H. (1989). A retrieval model incorporating hypertext links. In *Proceedings of the second annual ACM conference on Hypertext*, HYPERTEXT '89, pages 213–224, New York, NY, USA. ACM.

Dagum, P. and Luby, M. (1993). Approximating probabilistic inference in bayesian belief networks is np-hard. *Artif. Intell.*, 60:141–153.

Delbru, R., Rakhmawati, A. N., and Tummarello, G. (2010a). Sindice at semsearch 2010. Technical report.

Delbru, R., Toupikov, N., Catasta, M., Tummarello, G., and Decker, S. (2010b). Hierarchical link analysis for ranking web data. In *European Semantic Web Symposium / Conference*, pages 225–239.

Demartini, G., Iofciu, T., and de Vries, A. (2010). Overview of the inex 2009 entity ranking track. In *Focused Retrieval and Evaluation*, volume 6203 of *Lecture Notes in Computer Science*, pages 254–264. Springer Berlin / Heidelberg. 10.1007/978-3-642-14556-8-26.

Denoyer, L. and Gallinari, P. (2004). Bayesian network model for semi-structured document classification. *Inf. Process. Manage.*, 40(5):807–827.

Ding, D., Yang, J., Li, Q., Wang, L., and Wenyin, L. (2004a). Towards a flash search engine based on expressive semantics. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt. '04, pages 472–473, New York, NY, USA. ACM.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V., and Sachs, J. (2004b). Swoogle: a search and metadata engine for the semantic web. In

Proceedings of the 13th ACM international conference on Information and knowledge management, CIKM '04, pages 652–659, New York, NY, USA. ACM.

Ding, L., Finin, T., Joshi, A., Peng, Y., Pan, R., and Reddivari, P. (2005). Search on the semantic web. *Computer*, 38(10):62 – 69.

Domingue, J., Fensel, D., and Hendler, J. (2011). *Handbook of semantic web technologies*. Springer.

Dreyfus, S. E. and Wagner, R. A. (1971). The steiner problem in graphs. *Networks*, 1(3):195–207.

Elbassuoni, S., Ramanath, M., Schenkel, R., Sydow, M., and Weikum, G. (2009). Language-model-based ranking for queries on rdf-graphs. In *Proceeding of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 977–986, New York, NY, USA. ACM.

Fernandez, M., Lopez, V., Sabou, M., Uren, V., Vallet, D., Motta, E., and Castells, P. (2008). Semantic search meets the web. In *Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pages 253–260, Washington, DC, USA. IEEE Computer Society.

Florescu, D., Kossmann, D., and Manolescu, I. (2000). Integrating keyword search into xml query processing. *Computer Networks*, 33:119 – 135.

Fogaras, D. (2003). Where to start browsing the web. In *Innovative Internet Community Systems*, IICS, Third International Workshop, pages 65–79, Berlin Heidelberg. Springer-Verlag.

Fuhr, N. (1992). Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255.

Guha, R., McCool, R., and Miller, E. (2003). Semantic search. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 700–709, New York, NY, USA. ACM.

Guo, L., Shao, F., Botev, C., and Shanmugasundaram, J. (2003). Xrank: ranked keyword search over xml documents. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 16–27, New York, NY, USA. ACM.

Halpin, H., Herzig, D. M., Mika, P., Blanco, R., Pound, J., Thompson, H. S., and Duc, T. T. (2010). Evaluating ad-hoc object retrieval. In *Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010)*. 9th International Semantic Web Conference (ISWC2010).

Han, L. and Chen, G. (2006). The hws hybrid web search. *Information and Software Technology*, 48(8):687 – 695.

Harth, A., Hogan, A., Delbru, R., Umbrich, J., and Decker, S. (2007). Swse: answers before links. In *Proceedings of the Semantic Web Challenge*. CEUR-WS.org.

Harth, A., Kinsella, S., and Decker, S. (2009). Using naming authority to rank data and ontologies for web search. In *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pages 277–292, Berlin, Heidelberg. Springer-Verlag.

He, H., Wang, H., Yang, J., and Yu, P. S. (2007). Blinks: ranked keyword searches on graphs. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 305–316, New York, NY, USA. ACM.

Hildebrand, M., Ossenbruggen, J., and Hardman, L. (2007). An analysis of search-based user interaction on the semantic web. *CWI. Information Systems [INS]*, (INS-E0706).

Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., and Decker, S. (2011). Searching and browsing linked data with swse: the semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):365–401.

Hristidis, V. and Papakonstantinou, Y. (2002). Discover: keyword search in relational databases. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 670–681. VLDB Endowment.

Indrawan, M., Ghazfan, D., and Srinivasan, B. (1994). Using bayesian networks as retrieval engines. In *Proceedings of the 5th Australasian Conference on Information Systems*, ACIS, pages 259–271. TREC.

Jiang, X. and Tan, A. (2006). Ontosearch: a full-text search engine for the semantic web. In *Proceedings of the 21st national conference on Artificial intelligence*, volume 2, pages 1325–1330. AAAI Press.

Jones, K. S., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36:779–808.

Kacholia, V., Pandit, S., Chakrabarti, S., Sudarshan, S., Desai, R., and Karambelkar, H. (2005). Bidirectional expansion for keyword search on graph databases. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 505–516. VLDB Endowment.

Kasneci, G., Suchanek, F. M., Ifrim, G., Elbassuoni, S., Ramanath, M., and Weikum, G. (2008). Naga: harvesting, searching and ranking knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1285–1288, New York, NY, USA. ACM.

Kim, K.-M., Hong, J.-H., and Cho, S.-B. (2007). A semantic bayesian network approach

to retrieving information with intelligent conversational agents. *Information Processing and Management*, 43(1):225–236.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632.

Korb, K. B. and Nicholson, A. E. (2003). *Bayesian artificial intelligence*, volume 1. Chapman & Hall/CRC.

Koumenides, C., Alani, H., Shadbolt, N., and Salvadores, M. (2010). Global integration of public sector information. In *Web Science Conference 2010*, 26-27 April 2010, Raleigh, North Carolina.

Linckels, S., Repp, S., Karam, N., and Meinel, C. (2007). The virtual tele-task professor: semantic search in recorded lectures. In *Proceedings of the 38th SIGCSE technical symposium on Computer science education*, SIGCSE '07, pages 50–54, New York, NY, USA. ACM.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, 1 edition.

Myaeng, S. H., Jang, D.-H., Kim, M.-S., and Zhoo, Z.-C. (1998). A flexible model for retrieval of sgml documents. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 138–145, New York, NY, USA. ACM.

Nie, Z., Ma, Y., Shi, S., Wen, J.-R., and Ma, W.-Y. (2007). Web object retrieval. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 81–90, New York, NY, USA. ACM.

Nie, Z., Zhang, Y., Wen, J.-R., and Ma, W.-Y. (2005). Object-level ranking: bringing

order to web objects. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 567–574, New York, NY, USA. ACM.

Oniśko, A., Druzdzel, M., and Wasyluk, H. (2001). Learning bayesian network parameters from small data sets: application of noisy-or gates. *International Journal of Approximate Reasoning*, 27(2):165–182.

Oren, E., Delbru, R., Catasta, M., Cyganiak, R., Stenzhorn, H., and Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, 3(1):37–52.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report SIDL-WP-1999-0120, Stanford University.

Parthasarathy, K., Sreenivasa, K. P., and Dominic, D. (2011). Ranked answer graph construction for keyword queries on rdf graphs without distance neighbourhood restriction. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 361–366, New York, NY, USA. ACM.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge University Press, New York, NY, USA.

Pérez-Agüera, J. R., Arroyo, J., Greenberg, J., Iglesias, J. P., and Fresno, V. (2010). Using bm25f for semantic search. In *Proceedings of the 3rd International Semantic Search Workshop*, SEMSEARCH '10, pages 21–28, New York, NY, USA. ACM.

Porter, M. F. (2006). An algorithm for suffix stripping. *Program: electronic library and information systems*, 40(3):211–218.

Pound, J., Mika, P., and Zaragoza, H. (2010). Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 771–780, New York, NY, USA. ACM.

Rebane, G. and Pearl, J. (1987). The recovery of causal polytrees from statistical data. In *Proceedings of the 3rd Conference on Uncertainty in Artificial Intelligence*, UAI, pages 222–228, Amsterdam, NL. Elsevier.

Ribeiro-Neto, B. and Muntz, R. (1996). A belief network model for ir. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '96, pages 253–260, New York, NY, USA. ACM.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3:333–389.

Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple bm25 extension to multiple weighted fields. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, CIKM '04, pages 42–49, New York, NY, USA. ACM.

Rocha, C., Schwabe, D., and Aragao, M. P. (2004). A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 374–383, New York, NY, USA. ACM.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523.

Sanjay, A., Surajit, C., and Gautam, D. (2002). Dbxplorer: a system for keyword-based search over relational databases. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE, pages 5–16, Washington, DC, USA. IEEE Computer Society.

Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96 –101.

Sheth, A., Aleman-Meza, B., Arpinar, F. S., Sheth, A., Ramakrishnan, C., Bertram, C., Warke, Y., Anyanwu, K., Aleman-meza, B., Arpinar, I. B., , Kochut, K., Halaschek, C., Ramakrishnan, C., Warke, Y., Avant, D., Arpinar, F. S., Anyanwu, K., and Kochut, K. (2005). Semantic association identification and knowledge discovery for national security applications. *Journal of Database Management*, 16:33–53.

Sheth, A., Arpinar, I., and Kashyap, V. (2004). Relationships at the heart of semantic web: Modeling, discovering, and exploiting complex semantic relationships. *Enhancing the Power of the Internet*, 139:63–94.

Stojanovic, N., Studer, R., and Stojanovic, L. (2003). An approach for the ranking of query results in the semantic web. In *Proceedings of the 2nd International Semantic Web Conference*, ISWC '03, pages 500–516, Berlin, Heidelberg. Springer-Verlag.

Tran, T., Herzig, D. M., and Ladwig, G. (2011). Semsearchpro-using semantics throughout the search process. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):349–364.

Tran, T., Wang, H., Rudolph, S., and Cimiano, P. (2009). Top-k exploration of query candidates for efficient keyword search on graph-shaped (rdf) data. In *Proceedings of the 2009 IEEE International Conference on Data Engineering*, ICDE '09, pages 405–416, Washington, DC, USA. IEEE Computer Society.

Tummarello, G., Delbru, R., and Oren, E. (2007). Sindice.com: weaving the open linked data. In *Proceedings of the 6th International Semantic Web and 2nd Asian conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 552–565, Berlin, Heidelberg. Springer-Verlag.

Turtle, H. and Croft, W. B. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions in Information Systems*, 9:187–222.

Uren, V., Lei, Y., Lopez, V., Liu, H., Motta, E., and Giordanino, M. (2007). The usability of semantic search tools: A review. *The Knowledge Engineering Review*, 22(4):361–377.

Vallet-Weadon, D., Fernández-Sánchez, M., and Castells-Azpilicueta, P. (2005). The quest for information retrieval on the semantic web. *UPGRADE: the European Journal for the Informatics Professional. Monograph: The Semantic Web*, 2005(6):19–23.

Wei, W. and Barnaghi, P. M. (2007). Semantic support for medical image search and retrieval. In *Proceedings of the fifth IASTED International Conference: biomedical engineering*, BIEN '07, pages 315–319, Anaheim, CA, USA. ACTA Press.

Wu, G. and Li, J. (2007). Swrank: An approach for ranking semantic web reversely and consistently. In *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, pages 116–121, Washington, DC, USA. IEEE Computer Society.

Zhai, C. (2008). Statistical language models for information retrieval a critical review. *Foundations and Trends in Information Retrieval*, 2(3):137–213.

Zhou, Q., Wang, C., Xiong, M., Wang, H., and Yu, Y. (2007). Spark: Adapting keyword query to semantic search. In *Proceedings of the 6th International Semantic Web and 2nd Asian conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 694–707, Berlin, Heidelberg. Springer.

Zobel, J. and Moffat, A. (2006). Inverted files for text search engines. *ACM Computing Surveys (CSUR)*, 38(2):1–56.