Examining the Application of Modular and Contextualised Ontology in Query Expansions for Information Retrieval

by

David George, B.Sc. (Hons)

A thesis submitted in partial fulfilment for the requirements of the degree of Doctor of Philosophy at the University of Central Lancashire.

October 2010

Declaration

Concurrent registration for two or more academic awards

I declare that while registered as a candidate for the research degree, I have not been a registered candidate or enrolled student for another award of the University or other academic or professional institution.

Material submitted for another award

I declare that no material contained in the thesis has been used in any other submission for an academic award and is solely my own work.

Signature of Candidate: David George

Type of Award: Doctor of Philosophy

School: **Computing, Engineering and Physical Sciences**

Examining the Application of Modular and Contextualised Ontology in Query Expansions for Information Retrieval

Abstract

The purpose of this PhD is to use ontology-based query expansion (*OQE*) to improve search effectiveness by increasing search precision, i.e. retrieving relevant documents in the topmost ranked positions in a returned document list. Query experiments have required a novel search tool that can combine Semantic Web technologies in an otherwise traditional IR process using a Web document collection. The role of Ontology in the Semantic Web is to formally describe domains of interest and serve as contextual "anchors" to semantically retrieve and integrate information resources across the World Wide Web. However, an ontology can be monolithic or small and designed for shared or local use, so ontology reuse can be problematic because of design heterogeneity or partial overlap.

This research considers the ongoing challenge of semantics-based search from the perspective of how to exploit Semantic Web languages for search in the current Web environment. The research addresses two contributions to knowledge. The first concerns how modular, self-standing OWL ontologies (referred to later as contexts) could be employed in the prototype search tool. The second examines how the search tool could exploit Semantic Web-based *OQE* to improve information retrieval (IR) search effectiveness; this would be compared to traditional keyword-only search, on ordinary HTML documents. The primary objective has been to try to improve relevant document rankings (to increase precision). The return of additional relevant Web documents to improve recall, e.g. those containing none of the base query terms, would be a secondary benefit. Therefore, this research distinction is that Semantic Web technology would be applied to the traditional (unstructured/semi-structured) Web, as opposed to the Semantic (linked data) Web. An ancillary consideration will be how to facilitate reuse with minimal concept duplication (redundancy) and processing overhead, when ontology contexts are combined. Related to these issues will be how user interaction can be most effectively supported in the query process, to simplify selection of ontology contexts and their candidate *OQE* concepts.

A Java Jena-based semantic search tool, called SemSeT, has been developed to interrogate a large, independent TREC WT2g ¼ million Web document corpus by matching OWL file concepts with document text. Experiments have been conducted to identify keyword query expansion issues, through ontology traversal; in an attempt to demonstrate that ontology context-driven query expansion can improve IR precision, compared to traditional non-semantic search. This involved developing *OQE* algorithms and embedding a modified classic document relevance algorithm in the retrieval process, e.g. using a vector space model to increase the relevance weighting of relevant Web documents. A further task has been to examine the issue of semantic distance between *OQE* concepts and to identify appropriate concept relevance weightings to be applied the document ranking and retrieval algorithms. An approach has been developed to allow modular, self-standing OWL ontologies to be combined so that concept duplication (redundancy) and, therefore, processing overhead are minimised. Ontology contexts will themselves be used in a way that can help to guide a user in both selecting a query related ontology context and in identifying *OQE* terms when formulating queries.

The experiments will measure the success of *OQE* by comparing precision outcomes in the 10% to 30% recall range. Performance evaluation will be primarily based on an average of the precision percentage values for the 10%, 20% and 30% recall points (the APV). The experiments will show that a process combining next generation Semantic Web languages, *OQE* and ordinary Web document information retrieval, can exploit the benefits of ontology semantics in an otherwise traditional search environment, without resorting to indexing of RDF triple repositories and semantic reasoning-based RDF query languages.

Initial *OQE* experiments have had the effect of more than doubling APV performances and have maintained the differential up to 50% recall; further, extending *OQE* beyond a subsumption relationship, by exploiting the wider semantic relationships between ontology classes, has been fully justified, when using topic specific contexts. Some query results suggested that *OQE* may not be a solution to replace keyword-only search but could offer incremental search benefits in a bi-modal search process; however, subsequent modifications to concept relevance weights, involving higher weightings and even removal of weight differentials, have demonstrated that *OQE* can improve search precision by a further 10+% and that initial results could have been even more favourable.

Keywords:

Information Retrieval; Ontology Context; Ontology Reusability; Ontology-based Query Expansion; Precision and Recall; Semantic Search.

Contents

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of a number of people.

Firstly, I would like to thank to my supervisors, Zimin Wu (Director of Studies), Peter Gray and Roger Clowes for their support, guidance and contribution during my research. I must also thank Zimin and Peter for their regular participation and thought provoking discussions, which have been so helpful during my research journey; I am particularly grateful to Zimin for his direction and focus on key issues. Secondly, I am grateful to Helen Campbell, Janet Read and Nadia Chuzhanova for their helpful comments during my research.

During my daily work I have enjoyed the company of a friendly and lively group of fellow students and the support of the staff in the School of Computing, Engineering and Physical Sciences and the Science and Technology Graduate School.

Finally, I must thank my wife, Susan, for her patience, encouragement and support over the last few years.

INTRODUCTION

The purpose of this PhD is to use ontology-based query expansion (*OQE*) to improve search effectiveness by increasing search precision, i.e. retrieving relevant documents in the topmost ranked positions in a returned document list. Research experiments have required a novel search tool that can combine Semantic Web technologies in an otherwise traditional IR process using a Web document collection.

Growth in global interconnectivity has provided access to billions of information resources; often relying on simple keyword searches via search engines. However, keyword searches deliver only limited precision in identifying relevant documents and also may fail to identify relevant pages that contain related terms but none of the keywords. The retrieval challenge must inevitably progress to a semantic level, with users now requiring machine support to understand the contextual meaning of such diverse resources - through ontological underpinning, i.e. the true representation of a domain (Guarino, 1998, Wache et al., 2001). In a computing environment, ontologies are a formalised vocabulary of concepts, their relationships and explicit assumptions of a subject domain and represent an agreed "universe of discourse" that can serve as a reference point for related information sources.

Information search and retrieval is relatively straightforward on homogeneous sources but is more problematic when faced with semantic heterogeneity and information integration issues. These issues are potentially compromising when extracting meaningful and relevant information from autonomous and globally disparate but interconnected sources.

The Web has provided the platform for an "*information space of interrelated resources*" (W3C, 2004a) and the Semantic Web (Berners-Lee et al., 2001, Hendler et al., 2002) represents the next generation of the Web "*to create a universal medium for the exchange of data*". A number of issues have increased the profile of semantic interoperability, e.g. businesses have progressed from simply storing data to managing information and facilitating information retrieval (IR) and knowledge acquisition. Further, the need for improved retrieval of relevant data, by considering the semantics of the subject domain, is becoming increasingly important given the seemingly infinite volume of information on the Web.

Ontologies have featured in various academic search initiatives:

i. crawler-based locators of RDF and ontology resources, e.g. Swoogle (Ding et al., 2004) and Sindice (Oren et al., 2008); search support in specialist knowledge domains, e.g. bioinformatics and the Gene Ontology (Stevens et al., 2000, Ashburner et al., 2000);

ii. international organisation support, e.g. World Bank and Organisation for Economic Co-operation and Development (OECD) (Kim, 2005) and in legal document search

(Berrueta et al., 2006), where ontology query uses technical terms to find related information, terms and documents;

iii. other research involving word synsets, sense definition-based expansions and ontology-based query expansion (*OQE*): e.g. a review of *OQE* success factors (Bhogal et al., 2007); exploitation of ontological relations (Lei et al., 2006, Fang et al., 2005); word sense disambiguation in semantic network-based sense definitions (Navigli and Velardi, 2003); "hybrid" search combining ontology and keyword based IR results (Bhagdev et al., 2008) and earlier work on lexical-semantic query expansion work (Voorhees, 1994). Reasoning-based semantic query languages have featured in query expansions.

Commercial semantic search has included natural language processing search companies Hakia (Hakia, 2008) and Powerset (Powerset, 2008).

This research considers the ongoing challenge of semantics-based search and has similarities with research in (iii) above and addresses two contributions to knowledge. The first concerns how modular, self-standing OWL (W3C, 2004b) ontologies (to be termed contexts) could be used in *OQE*, in a bespoke semantic search tool termed *SemSeT*. The second examines how the search tool could manipulate such Semantic Web-based *OQE* to improve IR search effectiveness, compared to traditional keyword-only search, on unstructured HTML documents; i.e. as opposed to much of the above current research focus, of using semantic reasoning-based RDF query languages, on Semantic Web triple repositories, to refine the query process automatically. The primary objective is to try to improve relevant document rankings, to improve retrieval precision. The return of additional relevant Web documents to improve recall, e.g. those containing none of the base query terms, would be a secondary benefit. Therefore, the distinction is that Semantic Web technology would be applied to the traditional (unstructured/semi-structured) Web, as opposed to the Semantic (linked data) Web.

An ancillary consideration will be how to facilitate reuse with minimal concept duplication (redundancy) and processing overhead, when ontology contexts are combined. Related to these issues will be how the user can be assisted in the query process, i.e. to simplify selection of ontology contexts and their candidate *OQE* concepts.

A series of query experiments will identify the issues of keyword query expansion by ontology traversal; they will show that a process combining next generation Semantic Web languages, *OQE* and unstructured/semi-structured Web document information retrieval can exploit the benefits of ontology semantics in an otherwise traditional search environment. The experiments will assess the success of *OQE* against keyword-only search, by comparing precision outcomes, primarily in the 10% to 30% recall range. To provide a consistent approach, performance evaluation will be primarily based on an average of the precision percentage values for the 10%, 20% and 30% recall points (the APV).

The research will demonstrate that ontology context-driven query expansion can improve search effectiveness, compared to traditional non-semantic search, and the results will show that *OQE* can have the effect of more than doubling APV performances (in the 10% to 30% recall range) and can maintain the differential up to 50% recall. Later experiments with modified concept relevance weights, involving higher weightings and even removal of weight differentials, will demonstrate that *OQE* can improve search precision by a further 10+%, and that initial *OQE* results could have been even more favourable.

The remainder of this thesis is organised as follows. Chapter 1 will examine current developments, in both data and information integration and search activities, and present the research challenge and hypotheses. Chapter 2 will provide a high-level view of the research contribution tasks, i.e. proposed experimentation approach. Chapter 3 will discuss the experimentation search process, methods to be adopted, design work and implementation. Chapter 4 will present and analyse the experiment results and chapter 5 will summarise and evaluate the outcomes. Finally, chapter 6 will present an appraisal of the research method and its degree of success, together with an assessment of where future work should be directed.

1 LITERATURE REVIEW

This chapter will examine the issues that characterise the problem of integrating disparate, heterogeneous data and information systems, and documents, so that user search, by whatever mechanism, would be likely to return relevant information to a user. Related work will be considered, in terms of the significance and relevance of the work, and will include:

- a perspective on data, information and structural and semantic heterogeneity;
- data and information integration, interoperability and Web service;
- ontology principles, types and modelling, including Semantic Web languages and tools;
- information retrieval by search engine;
- modular ontology development;
- overall review and research challenge.

Whilst some of the areas may not appear to be directly related, they provide an evolutionary understanding of how a corporate and consumer society has contended with information integration and search issues. All the areas have relevance to the overall task of extracting meaningful and relevant information, from globally disparate but interconnected data sources, and they will provide the basis for guiding the discussion and justifying the selected research problem: i.e. how a semantics-based search tool might improve retrieval precision and recall using ontology-based query expansion.

1.1 A DATA AND INFORMATION PERSPECTIVE

Industry, commerce and society thirst on the need for information and this section considers the dynamics affecting communication and IR between organisations and individuals.

1.1.1 Dynamic Information Society

Organisations develop as a result of the complex demands of society and they survive by satisfying the needs of other organisations and customers; they have to handle technological development, aggressive market competition and expanding markets (Johnson McManus and Snyder, 2003). Such issues, compounded by business reorganisation and mergers driven by evolving corporate strategies, all stimulate organisational change - in the battle to stay ahead. The 21st Century workplace is a therefore a dynamic environment and many organisations demonstrate an insatiable need to reorganise and develop their information systems to understand markets, identify profitable customer segments, monitor performance, communicate and comply with government legislation (Rob and Coronel, 2002). Equally, financial constraints and profit maximisation, service or efficiency requirements, or the desire for strategic marketplace differentiation, all drive systems development programs and the challenge of integrating legacy and new information systems.

The success of effective organisation structures is determined by how well they meet the challenge of harmonising three key components of task, individuals and groups. Also, they achieve operational effectiveness by merged information extraction that supports communication and understanding by the information consumer.

1.1.2 Global Information Environment – Internet and Intranet

Many companies have gradually evolved as global organisations having data distributed in many parts of the world. Organisations have also attempted to achieve large-scale vertical integration with suppliers and customers, by transacting e-commerce through the Web. However, despite new database application development, organisations are often burdened by legacy database systems and consequently the need to retain and support associated applications (Stonebraker et al., 1993), and these can create fragmented information systems.

The Internet and, more specifically, the World Wide Web (Web) has provided the platform for a digital "information space of interrelated resources" (W3C, 2004a). The vitality and essential feature of the Web is its universality through its exploitation of the hypertext link; which makes it possible to link any document or data source to any other, in various environments: from the public or "open" Internet to corporate intranets and extranets.

Whilst public Internet sites tend to be open and not explicitly restricted to a particular class of users, intranets and extranets are more exclusive (Powell, 2002), e.g. an intranet is a shared information resource for employees, within a closed or discrete private network. Nevertheless, they employ standard Internet protocols (TCP/IP and HTTP) and Internet technologies (Bansler et al., 2000, Karlsbjerg and Damsgaard, 2001) and, whereas traditional client/server systems manage multiple applications and often have interface issues, intranet protocols use a common language and communicate via web-browsers that can access data held on different systems and stored in varied formats, thus providing a single, common graphical interface. Therefore, an organisation has the capability to instantly link geographically isolated operations with common, integrated, and up-to-date information. It is for such reasons that Web-based platforms have represented the platform for *emerging* data and communication technologies.

Recent Intranet/Extranet development, using Web-based information "portals", has shown that emerging technologies have been vital in supporting management philosophies that focus on changing organisation culture, e.g. promoting operational best-practise and employee empowerment to provide faster decisions and improved customer service, "openness" and sharing of information (Wagner et al., 2002, Bansler et al., 2000, Bar et al., 2000) and collaborative effort to harvest improved workforce productivity, e.g. consider the empirical study of US West (Bhattacherjee, 1998). The most productive intranets focus on news provision, enterprise-wide directory search facilities, and customised portal functionality (Lamb and Davidson, 2000); they generate widespread usage because end-users treat them as virtual libraries. However, their success depends on the integration of data sources.

IBM's "Dynamic Workplace" Intranet (Eliot and Barlow, 2002, Smeaton, 2002) has been attributed with revolutionising the way in which employees can communicate and access information. To reduce complexity, IBM's challenge was to merge more than 8,000 local intranets and link more than 11 million Web pages - to support 300,000 employees: *"there were far too many sources of information to search through ... key to our success ... was the goal of rendering the complexity of the organization irrelevant for employees"*.

1.1.3 Caught in a Web - the Price of Success

The seemingly inexorable penetration of the Internet and Web into daily life has unearthed retrieval problems because Web content is often stored in unstructured, natural language format. As a result, the current Web works well for creating and presenting different types of Web content but affords very limited support for meaningfully processing the data. This is because it is very much dependent on the human users for search, extraction, and interpretation activities.

The task of accessing information sources, ranging from unstructured and semi-structured text and data through to autonomous, federated and clustered database systems, can present users with potential information overload and the resultant problem of how to identify meaningful and relevant data. As a simple example, consider where different organisations post related information on the Web in different Web sites, in document form and dynamically via database access. However, whilst the information resources may be semantically related contextually, they are inevitably likely to be in varied formats; employing different terminology or data schema and therefore creating potential integration issues. Equally, consider a potential homebuyer seeking a certain range and type of property in an area with good employment prospects, low crime and highly rated schools and hospitals? In this case, to provide a comprehensive and meaningful answer, the data integration problem assumes different dimensions because a search could require access to autonomous databases holding say property, demographics, crime, health services and education data.

Such issues demonstrate the real world complexity that information systems must address and are consistent with the "Asilomar Report on Database Research" (Bernstein et al., 1998), which highlighted the need for the database community to radically address the way that technology captured, stored, analysed and presented the vast and increasing amount of online data. It was considered that the database community needed to widen its research to encompass all Web content and online databases, with a ten-year "Information Utility" goal: "*to make it easy for everyone to manage most human information online"*.

Clearly, the dramatic growth in the Internet and Web has brought with it the need for effective and flexible mechanisms to retrieve integrated and contextually related views from multiple information sources and data types; taking the homebuyer's use case, it requires a "mediation" of complex, multiple, real worlds that will support information and knowledge acquisition, which is increasingly and inevitably involving Web-based activities.

1.2 STRUCTURAL AND SEMANTIC HETEROGENEITY

Two issues play a significant role in creating disparities between information systems and repositories, namely organisational islands of development and differing designer influences in the developer process.

1.2.1 Development Autonomy

Development autonomy, or "islands of development", occurs where organisations have evolved as collections of distinct, autonomous departments with disconnected systems resulting from each pursuing their own IT infrastructure (Lamb and Davidson, 2000). An example of this was personally experienced during a career in financial services and banking, where mortgage, savings, unsecured lending, and insurance departments were historically allowed to develop autonomously - and specialised, heterogeneous systems were often *bought-in* to support new fast-track business strategies. Alternatively, development autonomy could occur simply because a database (DB) structure may be too complex to be modelled by one designer.

1.2.2 Design Autonomy

Design autonomy can be reflected in differing designer influence and choices in various areas: e.g. perception of the application/domain (universe of discourse), data model representation (model and query language), naming conventions, semantic interpretation of data, and constraints applied (Batini et al., 1986, Sheth and Larson, 1990, Bukhres et al., 1996). Thus, design autonomy produces differing perspectives, equivalence (but not identical) and incompatible design specifications.

Different perspectives can reflect different modelling and schema design, e.g. one schema S1 may show a relationship S1(Employee:Dept) compared to another schema S2 showing S2(Employee:Project:Dept), or a name inconsistency between related entities or attributes. Equivalence among model constructs exists when different constructs are used to model the concept equivalently e.g. where entities in one schema are modelled as attributes in another or where there are generalisation or specialisation differences e.g. in object class hierarchies – as will be seen later in subsection 1.2.4.

Finally, incompatible design specifications result in conflict, e.g. by specification of different data types, cardinality or referential integrity.

1.2.3 Modelling the Real World

Semantic heterogeneities represent differences in the real world interpretation of subject context and meaning of data, e.g. which often occurs during a database designer's task of translating conceptualisations of the real world into the representational world of DBs - see Fig. 1.

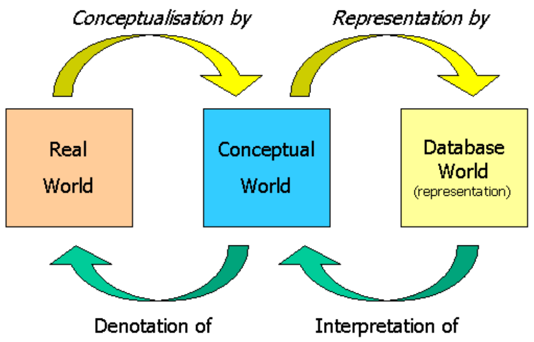


Fig. 1. Relationship between real, conceptual and representational (DB) worlds.

They reflect data model, schema construct, and data inconsistencies in the conceptual and DB worlds (Kim et al., 1993, Hammer and McLeod, 1993, Kashyap and Sheth, 1996, Garcia-Solaco et al., 1996). Where two objects represent the same *concept* (of the entity or object) there may be a semantic relationship, or equivalence, but if the *contexts* (i.e. the universes of discourse) differ, e.g. when analysing employee data across two different companies, then different extensions will result, e.g. different instances of employee. Conversely, where extensions are the same in two entities they may be semantically unrelated e.g. two identical groups of people but one group happens to represent an operational department and one a project team. Semantic understanding is based on the relationship between concept and context, and the identification of semantic heterogeneity requires consideration of both such issues. As will be seen later, semantic heterogeneity is both prevalent and a cause of semantic conflict in all technologies applied to data, information and knowledge representation linking autonomous operations.

1.2.4 Heterogeneity Resulting from Autonomy

In an analysis of schema integration methodologies (Batini et al., 1986), structural and semantic heterogeneity categories were specified as those involving *naming* conflicts and those involving *structural* conflicts.

Naming conflicts occur when different terminology is used across organisations. Differences in entity or attribute naming are classified as either *homonyms* (differing concepts but having same name) or *synonyms* (same concepts but having different names). Structural conflicts occur when a different choice of modelling construct is employed, e.g. Fig. 2 shows how equivalent person constructs can be represented: either in a generalisation *hierarchy*, e.g. where one schema contains a general entity or class (*hypernym*) Person with differentiating specialisation entity (*hyponym*) types Female and Male, or where another schema may collectively represent all persons within the generalisation entity Person, with any person classification represented via an attribute like Gender. Thus we can see that the concept Female would be explicitly represented as a Female entity in one schema but only implicitly represented, i.e. as an entity by the value "Female" in the Gender attribute in the other.

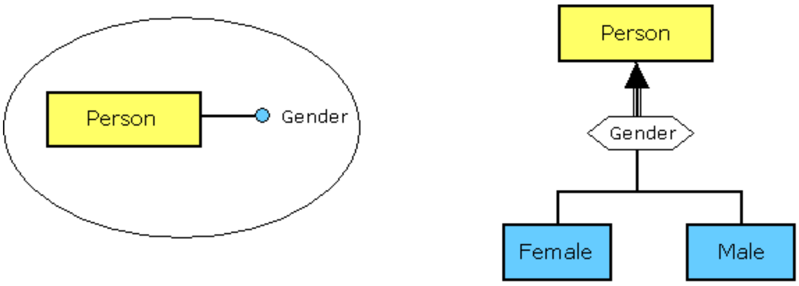


Fig. 2. Structural conflict in representation of Person entity.

Such issues were also recognised by a study of heterogeneity in federated DB systems (Hammer and McLeod, 1993), which referred to differences in: *metadata specification* of the conceptual schema (conflicts in structure of relationships) and *object comparability* (e.g. in naming through synonyms and homonyms). Similarly, a wider-ranging classification of heterogeneities (Kim et al., 1993) examined structural conflicts based on integrations of entity-relationship (E-R) and object-oriented (O-O) schemas and identified two key causes of semantic conflict: where component schemas use different *structures* to represent the same information, e.g. *entity structure* conflicts, through missing attributes (differences in number of attributes), and where different *specifications* are used for related or similar structures, e.g. *entity name* conflicts evidenced by different names for equivalent entities (synonym) or same name for different entities (homonym). Also considered were entity attribute conflicts caused when one schema uses an entity and another uses an attribute to represent the same information. A comparison of the taxonomy with that of (Kashyap and Sheth, 1996) shows similar conflict classifications.

The above studies appear to have been effectively subsumed in a comprehensive taxonomy of issues relating to multidatabases (Garcia-Solaco et al., 1996), which sought to provide a concise explanation of conflicts based on O-O components of object classes, class structures, and object instances; from the E-R perspective, a class can be compared to a table and an object to a record. The study focused on two particular distinctions:

- semantic heterogeneities between object classes: including (i) differences in *names* such as involving class and attribute *synonymy* of names (e.g. where one schema may refer to customer whereas another may refer to client) and homonymy, or *polysemy* (e.g. where an attribute market might relate to product or customer in different schemas); or (ii) differences between attributes, e.g. temporal conflicts (such as employee role: past vs. present); or (iii) attribute domain differences (e.g. unit of measure and scale conflicts).
- semantic heterogeneities between class structures: including (i) *generalisation* and *specialisation* inconsistencies, reflecting heterogeneities between classification of super-class and sub-classes: e.g. employees specialised as male and female groups vs.

occupation groups), or (ii) *aggregation* and *composition* conflicts: e.g. where seemingly similar object classes might actually be represented by differing collections of object classes - such as Person(address, tel.) in one database versus Person(street, city, county, tel.) in another.

Whilst these classifications represent just a small part of the semantic conflict taxonomy they serve to underline the difficulties that information query and retrieval systems can encounter when processing and interrogating data and information.

1.3 DATA AND INFORMATION INTEGRATION

The last 30 plus years have witnessed two paradigms in the data integration challenge - the development of the E-R and O-O models (Chen, 1976, Kim, 1991). In the last quarter century, data integration has been a key issue in achieving systems interoperability between heterogeneous data storage and management systems because of the existence of system, schema, and semantic heterogeneity.

Whilst DB technology has in the past had a significant impact on this problem, the exponential growth in diverse information accessible via the Web has made IR increasingly complex, with billions of documents being accessed by over 300 million users (Patel-Schneider and Fensel, 2002). The combination of structured DB resources, and semi-structured and unstructured Web data, has resulted in systems interoperability and online-data integration representing some of the most significant challenges facing the information technology (IT) community in the last 25 years; with the cost of data integration and improving data quality estimated at $1bn a year (Brodie, 2003).

Integration can be achieved by addressing the interoperability dimensions of distribution, autonomy and heterogeneity (Sheth, 1998). This problem has received considerable interest from researchers in the DB and artificial intelligence fields (Levy, 1999), and has resulted in three generations of information systems interoperability evolution: the period to the mid-eighties, the period to the mid-nineties, and the mid-nineties onwards.

1.3.1 Evolution of Interoperability Initiatives

The objective of data and information integration is to provide a uniform interface to a variety of disparate and distributed data source types that demonstrate heterogeneity. Firstly, source heterogeneity is evidenced in structured data: relational, extended-relational, and object-oriented DBs where schema and data are separated and structural consistency of records in schema objects is implicit in the design. Secondly, it is evidenced in semi-structured data: as in HTML and XML documents, where there is no guarantee of consistency of data structure or requirement for a pre-defined schema to which data objects must conform. XML is sometimes called self-describing data stored within its own structure (Elmasri and Navathe, 2004).

Thirdly, it is found in unstructured data: represented by text files, images including MRI scans and X-Rays, audio, and video - all of which have no schema at all.

The three evolutionary periods of development are portrayed in Fig. 3. In the first generation, organisations were characterised by having large volumes of departmental data, yet needing to share data between departments. The DB integration problem manifested itself with the development of multidatabase systems (Batini et al., 1986), where the emphasis was on system and data management as opposed to information or knowledge management. However, changes in approaches were driven not only by the need to integrate heterogeneous DB systems (Sheth and Larson, 1990, Drew et al., 1993, Bright, 1994), where the solution involved the development of federated database systems (FDBS), but also by the need to integrate heterogeneous data stored in a variety of forms (Wiederhold, 1999).

Second generation interoperability became more focused towards structure (data schema) and syntax (data types) than systems, and on wider-scale network distributions that showed increasing evidence of object-orientation. With the expansion of the Web, second-generation integration initiatives witnessed the development of federated *information systems* that addressed both structured DBs and the wider range of semi-structured and unstructured data sources. These systems included mediator/wrapper architectures that generate a mediated schema as a *homogeneous* and *virtual* information source, without integrating the data resources, and other online information systems making more extensive use of metadata (Wiederhold, 1992, Levy et al., 1996, Garcia-Molina et al., 1997, Bertino et al., 2001). Metadata (data about data) encompassed a variety of forms beyond simply schema, including DB descriptions, content descriptions of images and audio, and HTML/SGML document type definitions.

In the third generation, the phenomenal expansion of the Internet and e-business has resulted in growth in the volume and types of information, with increasing exploitation of XML-based languages. It has also created the need to effectively integrate information repositories, such as in content management of digital libraries, application integration via workflow systems and messaging, and data mining and on-line analytical processing for business intelligence (Roth et al., 2002). Global interconnectivity resulted in the emerging *global information infrastructure* (GII) (Kashyap and Sheth, 2000). However, whilst providing access to billions of information resources, access to meaningful and relevant data often relied (and in many ways still does) on simple keyword searches via search engines (Gudivada et al., 1997). However, as keyword searches deliver only limited precision in identifying relevant information, the main challenge has progressed to a semantic level, i.e. requiring machine support that functions in a cooperative and collaborative way to understand the contexts of such diverse resources through metadata.

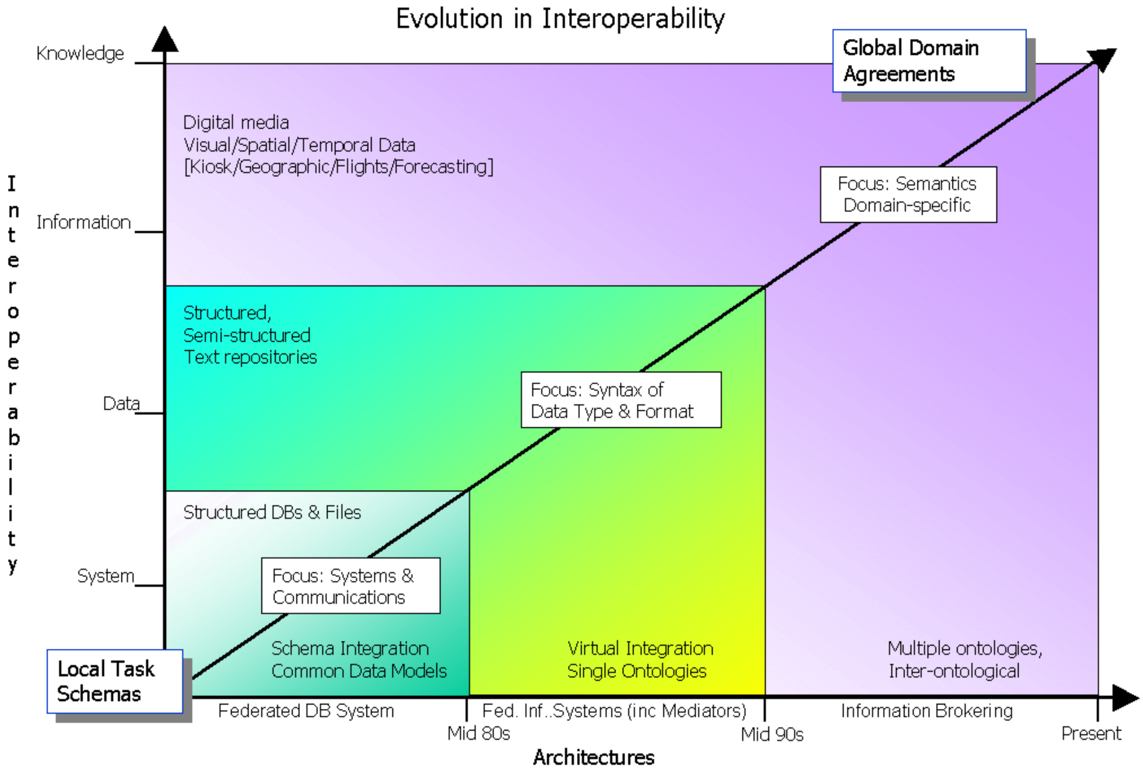


Fig. 3. Evolution in integration and interoperability.

Cooperative information systems focused on interactivity between autonomous components and such systems gained prominence during the 90s (De Michelis et al., 1997, Klusch, 2001). They provided methods and tools to access large amounts of information, computing services, and support individual or collaborative human work. Multi-agent systems, using intelligent information agents (Knoblock et al., 1994), provided a solution for supporting information brokering systems that were supported by vocabularies. The shift from managing data and information, to knowledge acquisition, resulted in the need for greater semantic interoperability. Enterprise and global information systems (GIS) domains required content and representation of information to be more closely related to domain specific concepts, enabled through metadata and shared ontologies (Gruber, 1993, Guarino, 1998). The predominant architectures were multi-modal information brokering systems (Ouksel and Sheth, 1999, Bergamaschi et al., 1999), using semantics described by potentially multiple ontologies (de Bruijn, 2003) and the support of artificial intelligence (AI) for information queries.

Clearly, the scale of the integration challenge is changing, requiring the database community to widen its research to encompass all Web content and online databases; thus interoperation is key to making it easier for everyone to manage most human information online (Bernstein et al., 1998, Gray et al., 2000). The paradigm of collaborative intelligent agents (Knoblock et al., 1994), searching for metadata qualified information in Information Brokering and Web Services, inevitably invites consideration of how ontologies could be exploited in semantics-based search particularly in view of the emerging Semantic Web. This will be considered in more detail later.

1.3.2 Integration and Interoperation

At this stage, it is appropriate to make a distinction between integration and interoperation (Wiederhold, 1999).

FDBSs enable scalable integration, and provide a balance between shared data *integration* and federated user autonomy. Component DB autonomy is secure as schema and data management remains under local control, and data sharing relies on each local database administrator to define the data schema subset elements to be made available to the federated system users (Parent and Spaccapietra, 2000). So, FDBS users share a common, *static* schema that provides search functionality across the distributed federation component systems; any search results in effect mirror the *pre-defined* schema views accessed via the user application, and would depend on the complexity of query developed by the user. This can be viewed as representing a basic data and information integration approach, where DB source views are in effect combined (or fused). As a generalisation, it is little different to queries of any DB system.

In comparison, mediator based information integration through *interoperation* across diverse data sources is a different and more *dynamic* way of increasing the value of information by abstracting information from disparate data sources on a selective basis, e.g. a travel system might combine airline flight, hotel chain, insurance, and airport car park and tourist excursions, stored in related but essentially domain specific and autonomous systems. In these mediator-wrapper and information brokering systems, user applications deal with higher-level query aspects while query-planning, selection and summarisation are separate, i.e. they are left to intelligent mediators, wrappers and agents, where mediators integrate data from multiple sources provided by other mediators, agents and source translators. Therefore, in this sense, integration by interoperation represents a more dynamic and *flexible* or *cooperative* approach.

1.3.3 Schema versus Ontology

During the literature review it became evident that the terms *schema*, *integration*, and *ontology* have been regularly used in the same data and information context, even though there is a difference between schema and ontology; a broad perspective is offered on this issue.

In the simplest case, DB schema modelling usually defines the structure and integrity of data elements in a single “enterprise” application - although not necessarily in a single DB. Therefore, the development of data models invariably supports just the specific needs and activities of the particular organisation. Any semantics described in data models are therefore *local*, i.e. they can be considered to represent an *informal* agreement between a developer and department users in that unique or singular environment. However, ontology structures differ because the fundamental principle of a computing ontology is the *formal* representation of generic knowledge through an *agreed* logical view of the domain of interest, i.e. an ontology describes the domain with a *global* view; because it has more relevance as domain classification and tends not to be task specific. These characteristics can be represented at various levels in how a hierarchy of data, information and knowledge "integration" approaches could be perceived - as depicted in Fig. 4.

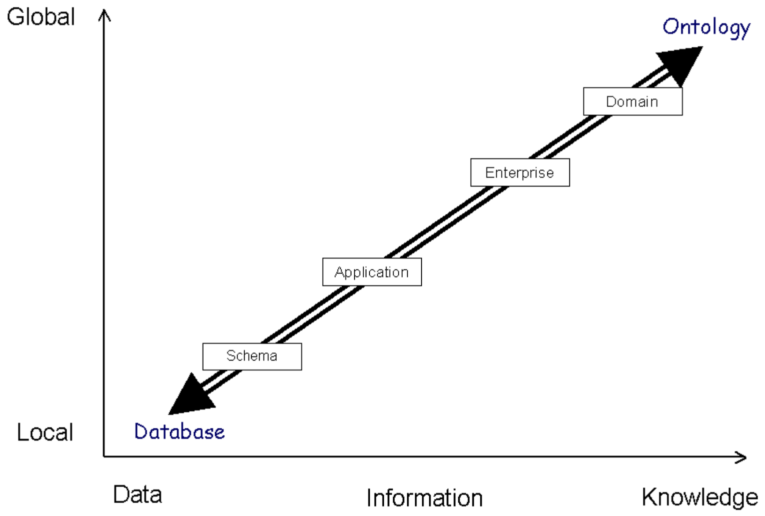


Fig. 4. Hierarchy of data, information and knowledge integration.

Equally, it can be said that traditional data integration, by global schema, represents a retroactive and maintenance approach, e.g. to merge two or more existing schema and to remove semantic heterogeneity; whereas an ontological integration approach is driven from the perspective of knowledge sharing, through formalised semantics, and can act as the precursor and foundation for semantic integration. For example, a *general* ontology can operate as a standard on which future specialised domain-specific ontologies can be aligned. Hence ontology offers a top-down, proactive approach and schema integration a bottom-up, retroactive solution.

An Ontology may appear to have a similar function to a DB schema, but the key differences have been succinctly described (Horrocks et al., 2000):

- the definition (specification) of ontologies requires a language syntactically and semantically more expressive than languages used in DBs;
- as ontology provides a domain theory used for information sharing and exchange, it must therefore it must equally use a shared and consensual terminology;
- unlike a DB, an ontology is a structure to represent knowledge - not to contain data.

The ontology super and sub class (subsumption) hierarchy represents a generalisation and specialisation of concepts; providing parallels with hypernym and hyponym in DBs.

1.3.4 Web-based Information and Service Integration

Regardless of the issues of local and global, and informal and formal integration mechanisms, the key issue has become the need to provide global access to DBs and knowledge bases (KBs) for information search, using Web search tools.

Data and information search is no longer restricted to organisational need but is required by the global community that is the Internet. Therefore, a sophisticated Web search facility that can interpret data and information sources is becoming increasingly relevant, regardless of the structural and semantic heterogeneity characteristics of data storage and information/knowledge representation approaches. However, the current approaches of commercial search engines offers a less formalised and semantically weak method of dynamically extracting, integrating and presenting lists of heterogeneous data and information sources that may or may not be potentially relevant. As will be discussed later, there is little commercial evidence that formal knowledge structures are being used in their processes to achieve semantic precision/integration in retrieved document hit lists. This could be improved if greater weighting could be applied to documents that contain contextually related terms matching some ontological description of the query domain, i.e. using the vocabulary of an ontology to expand a search query.

The task of accessing billions of information sources ranging from unstructured and semi-structured, and structured data presents users with the problem of how to identify relevant data. Most knowledge on the Web is in natural language, unstructured text, often supported by graphics, which may be convenient for human understanding but is difficult for machine interpretation. This is because natural text restricts the indexing capabilities of search engines, as they cannot infer meaning (Ding et al., 2005). Next-generation technologies are now being developed to address these challenges, such as Web Services (McIlraith et al., 2001, Brodie, 2002, Sycara et al., 2004) and Semantic Web (Berners-Lee et al., 2001, Hendler et al., 2002).

In Web Services, the traditional concept of the Web, being designed for human interpretation and solely a repository for text and images, is now being utilised as an integrated “provider of services”; where a typical service operation, e.g. offering holiday and flight-bookings, would use tools to build “virtual” advanced systems accessing multiple distributed systems supplied by different organisations.

The Semantic Web is said to represent the next generation of the Web, with the objective of creating a universal medium for the exchange of data, information and knowledge by representing it in a standardised data description language and linking it to formalised vocabularies defined in ontologies. Focus has therefore logically moved towards understanding how Ontology-based structures can link disparate data sources and provide intelligent search functionality. However, the Semantic Web is not currently particularly high profile in Web search activities.

Standardisation at different layers of information systems architectures is important and, as will be discussed later, several key enabling technologies have been adopted as World Wide Web Consortium (W3C) recommendations: the Resource Description Framework (RDF) core language (W3C, 2004c), and the RDF Schema and OWL Web Ontology languages (W3C, 2004b), all constructed using the universal XML syntax.

1.4 LINKING AND SHARING INFORMATION BY ONTOLOGY

Ontologies are used to capture knowledge about a domain of interest by describing concepts (classes), relationships (properties) between those concepts, and constraints (restrictions) that may be specified on relationships. As previously mentioned, ontology structures differ from database schema because the fundamental principle of a computing ontology is the formalised representation of knowledge agreed for sharing, in a language that provides a logical view of a subject area or domain. This is achieved through an accepted vocabulary and definition of the member concepts and their relationships; that can be re-used by different applications (Spyns et al., 2002, Noy and Klein, 2002) e.g. operating in the context of open environments such as the Semantic Web.

Therefore, compared to database schema there is a greater *formality* in the way in which ontologies represent knowledge for a community of users, because ontologies are always intended to be a true representation of a domain (Guarino, 1998). As already shown in Fig. 4, ontologies and data models are appropriate at different levels of task-specificity, with ontologies being more generic and task-independent (Kalinichenko et al., 2003).

1.4.1 Ontology Theory

In the context of knowledge sharing in computing, ontology is a formal vocabulary representing concepts and relationships in an application area; therefore ontology represents a "universe of discourse" to which Web contents can refer. Ontologies enumerate, or detail, concepts and their attributes, the relationships between concepts, and any constraints on those relationships. The term "Ontology" is derived from Greek philosophy, via the terms "Onto" (being or existence) and "logia" (written or spoken discourse).

A widely cited definition of an ontology has been provided by Gruber and subsequently modified by Borst (Gruber, 1993, Gruber, 1995, Borst, 1997):

> *"an Ontology is a formal, explicit specification of a shared conceptualization"*

In this statement, the type of concepts used and the constraints on their use, are "explicitly" defined and "formal" implies that the ontology specification should be machine-readable. The term "shared" reflects that ontology should capture consensual knowledge or commitment by the communities. Interestingly, Gruber also says that:

> *"a commitment to a common ontology is a guarantee of consistency, but not completeness"*

This was further refined (Guarino, 1998) by defining an ontology as containing:

> *"a set of logical axioms designed to account for the intended meaning of a vocabulary"*

The fact that humans are able to readily abstract information from, e.g. sounds, images and video, illustrates that representation at a higher semantic level reinforces the correlation between data and information. Therefore, it can be argued that effectively answering user queries of heterogeneous digital data types demands information, or semantic-level, correlation to improve query response precision. This can be achieved by describing information at three levels (Kashyap et al., 1995, Kashyap and Sheth, 1996, Arch-Int and Sophatsathit, 2003), i.e. as shown in the ontology, metadata and data levels depicted in Fig. 5.

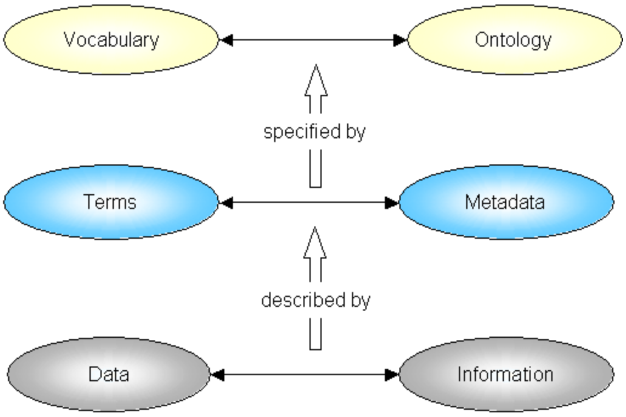


Fig. 5. The relationship between data, metadata and ontology.

Metadata provides information about data and information resources, i.e. it summarises information content to provide a metadata context. However, if a semantic correlation is to be achieved, metadata must also convey the meaning of data. Data and information sources will often contain a set of meta terms, in the form of keywords to represent the abstracted vocabulary of the content. Similarly, Semantic Web resources are described using metadata annotations and these can be determined, or specified, by a contextually relevant ontology formalised in an ontology language; this ontology will specify concepts (classes) and their roles (relations), and provide semantic anchors to give meaning to data on the Web. In addition, because ontologies represent *shared* specifications, they can be used for the annotation and linking of multiple data and information resources.

1.4.2 Types of Ontology

The main types of ontologies can be represented at the three levels of granularity depicted in Fig. 6, i.e. *top-level* ontologies, *domain* and *task* ontologies, and *application-level* ontologies (Guarino, 1998). In effect, these represent the degree of accuracy that can be achieved in characterising the *conceptualisation* (Gruber, 1993, Gruber, 1995, Borst, 1997) that they *formalise*. A course-grained or top-level ontology represents a generalised, imprecise, more abstract structure that therefore becomes more *shareable* to a wider range of domains and applications; but equally becomes less supportive at the domain and application levels because of limited expressivity. On the other hand, a fine-grained ontology represents a more precise, specialised, and real specification; one that may be more domain and application supportive but, at the same time, less shareable.

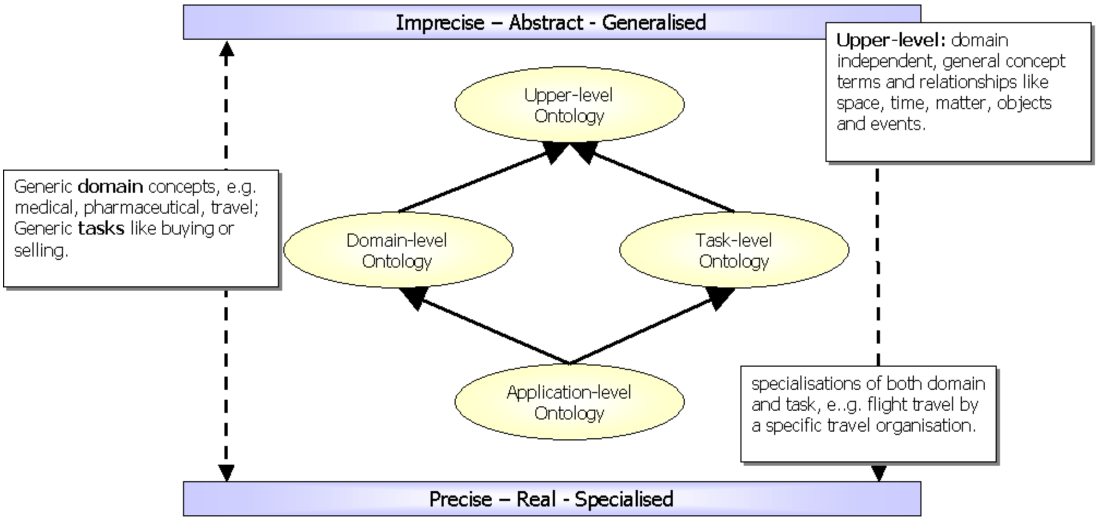

Imprecise – Abstract - Generalised
Upper-level: domain independent, general concept terms and relationships like space, time, matter, objects and events.
Upper-level Ontology
Generic domain concepts, e.g. medical, pharmaceutical, travel; Generic tasks like buying or selling.
Domain-level Ontology
Task-level Ontology
Application-level Ontology
specialisations of both domain and task, e.g. flight travel by a specific travel organisation.
Precise – Real - Specialised

Fig. 6. Ontology type classification (Guarino, 1998).

Top-level ontologies tend to describe abstract general concept terms and relationships like space, time, matter, objects and events, and are domain-independent. Examples are WordNet (Fellbaum, 1998), a lexical database resource for natural language processing systems (language, speech and communication), SUMO (Niles and Pease, 2001) defining general-purpose terms to act as a foundation for specific domain ontologies, and the knowledge or commonsense-based Cyc ontology (Lenat, 1995) that is reflected in OpenCyc (Cycorp, 2005). A useful early survey of ontologies was presented in “The State of the Art in Ontology Design” (Noy and Hafner, 1997).

Domain and task ontologies, respectively, provide vocabularies about concepts in their generic domains (e.g. medical, pharmaceutical, computing or travel), or generic tasks (e.g. buying or selling). Application ontologies define concepts that are application dependent, i.e. are specialisations of both domain and task, e.g. related to flight travel by a specific airline partnership, or purchasing in stock-market activities as opposed to shopping.

1.4.3 Ontology Expressiveness

The constructors, or resources, to formally specify ontology may be founded on either existing ontologies or formal classification systems. Examinations of the different types of ontology have revealed a “spectrum” of *ontology expressiveness* (de Bruijn, 2003) and a comparison based on categorisations of ontologies can be determined by the information an ontology needs to convey (Lassila and McGuinness, 2001, Uschold and Gruninger, 2004).

Based on the above, a modified comparison is presented in Fig. 7 to show the spectrum differentiated in two dimensions: by distinguishing formalism by the degree of *specification* along the y-axis and by separating the type, or purpose, of the ontology along the x-axis, i.e. recognising the degree of generalisation or specialisation as discussed in subsection 1.4.2. The comparison charts the level of formalism (ranging from a simple term list to complex and highly descriptive and constrained ontology) against the granularity/type of the ontology (i.e. by either application specific, or generic domain and task ontology, or abstract specifications).

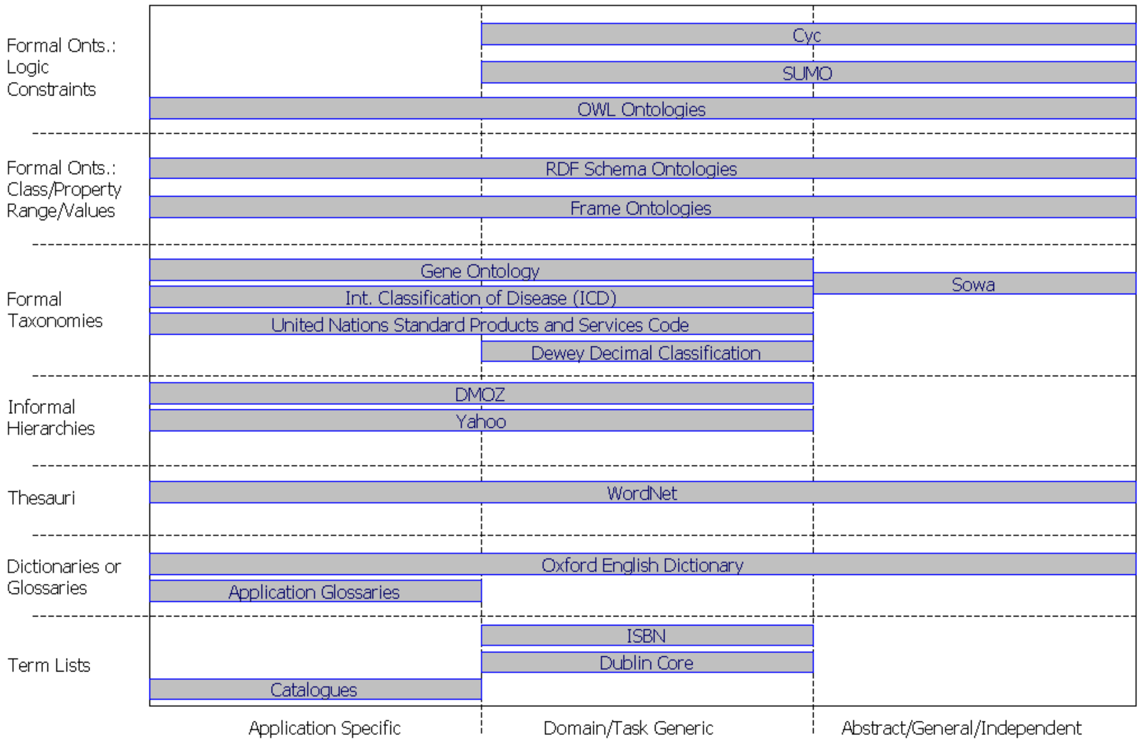


Fig. 7. Analysis of expressiveness by ontology type.

Considering the y-axis, the least formalised levels are represented by controlled vocabularies such as a list of terms or catalogues; more detailed examples would be technical glossaries, providing explanation for terms, and dictionaries.

- The Dublin Core represents a standard for cross-domain information resource description. It contains 15-element metadata set, used in digital libraries, with the objective of facilitating discovery of electronic resources.
- The International Standard Book Number (ISBN) is a unique numeric commercial book identifier and is based upon a 10-13 digit code.
- The Oxford English Dictionary contains English-spoken words from across the world and is recognised as the authority on the evolution of the English language.

The next logical group includes thesauri, where additional semantics between terms are specified (e.g. synonym and hypernym relationships), and informal hierarchies/taxonomies, where an explicit hierarchy of generalisation and specialisation is supported but without strict inheritance being implied, e.g. where a the terms Internet and Software might be considered relevant to the concept of Computers but could not be logically defined as sub-classes.

- WordNet® is a large lexical database providing a thesaurus of all English language words. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets). Synsets are organised by lexical semantic relations, e.g. for nouns and verbs, they include hypernym (*is-a* relation), hyponyms, synonyms, meronyms and antonyms, together with sense definitions. WordNet® is connected to various Semantic Web databases and is commonly re-used via mappings to concepts in other ontologies.
- Informal *is-a* hierarchies can be found in the directory classifications of Web sites, e.g. the DMOZ Open Directory Project (DMOZ, 2008) is the largest, most comprehensive human-edited directory of the Web and is used in many search engine processes. The Yahoo Directory [1], managed by staff, has a similar role.

The next level, formal taxonomy (hierarchical classification), display strict inheritance**.**

- The Dewey Decimal Classification (DDC) is a library classification system.
- The United Nations Standard Products and Services Code (UNSPSC, 1998) provides an open, global multi-sector standard for classification of products and services and has a hierarchical classification with five levels.
- The International Classification of Diseases (ICD) [2], published by the World Health Organisation, is the international standard used for health management and clinical use; it analyses and classifies diseases and health problems, including the general state of health of population groups.
- The Gene Ontology (GO) project (Ashburner et al., 2000) is a collaborative effort to unify the representation of gene and gene product attributes and is part of a larger classification, the Open Biomedical Ontologies (OBO).
- SOWA's top-level lattice structure ontology has 27 concepts derived from logic, linguistics, philosophy and artificial intelligence.

In varying degrees, all of the above levels can be viewed as lightweight ontologies.

Lastly, the higher levels contain the more expressive, heavyweight ontologies, of which the first group are characterised by the use of ontology languages that are able to express any specified range and value *restrictions*, i.e. where values of properties are restricted, e.g. by data type, or where general logic constraints are applied to values by logical or mathematical formulas using values from other properties.

[1] Yahoo: http://dir.yahoo.com/.

[2] ICD: http://www3.who.int/icd/vol1htm2003/fr-icd.htm.

The first group includes.

- The Ontolingua ontology-building language is based on KIF and Frame Ontology defined terms to represent objects; a frame, or class, contains a number of properties that are inherited by subclasses and instances.
- The Semantic Web RDF Schema is a W3C recommendation ontology language, i.e. it is a vocabulary for describing classes and properties of RDF-based resources, with semantics for generalised hierarchies. RDF is based on the extensible markup language (XML) and is in effect a subset of OWL, i.e. all RDF Schema is OWL.

The second group contains ontologies described using very expressive ontology languages including using First-Order Logic constraints, where there are not only constraints between terms but more detailed relationships, e.g. disjoint classes, inverse relationships, part-whole relationships.

- The W3C recommendation Web Ontology Language (OWL) is a vocabulary extension of RDF Schema and provides greater machine interpretability of Web content than RDF Schema; particularly the more expressive OWL DL and Full species.
- The Suggested Upper Merged Ontology (SUMO) is a First-Order Logic upper and foundation ontology for information processing systems that has been expanded to include a mid-level ontology and various domain ontologies.
- The Cyc Knowledge Server is a very large, formalised and highly expressive, multi-contextual knowledge base and inference engine developed by Cycorp. Cyc describes fundamental human knowledge: facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life.

1.4.4 Ontology Modelling Approaches

Whilst the function of an ontology is to represent knowledge through the use of formal semantics, different types of ontology can be modelled with different types of modelling techniques and languages. Early knowledge representation (KR) was based on structured inheritance networks (Nardi and Brachman, 2002) that displayed networked semantics.

In the early 90's ontologies were formalised using Frames and First-Order Logic, e.g. Cyc (Lenat, 1995), Ontolingua (Gruber, 1992). More recently, Description Logics (DL) have prevailed (Baader et al., 2003), resulting in DL languages, e.g. OIL (Horrocks et al., 2000, Fensel et al., 2001), DAML+OIL (Horrocks, 2002), and OWL (W3C, 2004b).

It is clear is that a variety of approaches involving: modelling terminologies (e.g. concepts and roles, classes and relations), KR paradigms (e.g. Frames, DL) and implementation languages, have been developed. As will be seen later, OWL and ontology development tools like Protégé (Noy et al., 2001) demonstrate fundamental relationships with Frames and Description Logic.

Semantic Networks

One of the oldest knowledge representation formalisms is semantic (or inheritance) networks. Semantic networks represent a generic network, in that they avoid references to any particular system (Nardi and Brachman, 2002). In this formalism, a node in a graph characterises either a concept or concept *member* and arcs connect concepts/members that are semantically related. Meaning is conveyed by the way a concept is connected to other concepts. This is demonstrated in Fig. 8, which contains some nodes representing atomic (self-standing) concepts, e.g. Building, and arcs (links), characterising relationships between concepts. Two main types of arc are shown, i.e. an *is-a* arc and an *instance-of* arc. An *is-a* arc indicates that one class (e.g. City) is a sub-class of another (i.e. Pop_Centre) and this represents a basis for inheritance, i.e. either denoted as a "subset" relation City $\sqsubseteq$ Pop_Centre or "superset" relation Pop_Centre $\sqsupseteq$ City. An *instance-of* defines that a concept member, e.g. "Leeds", is an example of another concept City, i.e. it displays an "element of" relation.

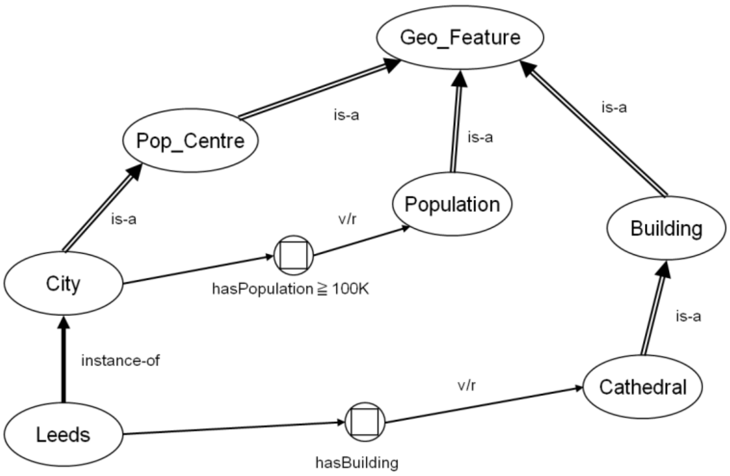


Fig. 8. Example of a Semantic Network.

Concepts denote what are termed unary descriptions, e.g. City $\sqsubseteq$ Pop_Centre, whereas links denote binary relationships, e.g. hasBuilding(X, Y). The set of *is-a* arcs thus specifies a subsumption order, or hierarchy, on classes (i.e. one is subsumed by the other); this is often termed a taxonomy or classification hierarchy. Therefore, a semantic network structure can be used to either generalise a concept, by employing a more abstract definition, or increasingly specialise a concept to a more specific class level. The overall structure can be referred to as a terminology.

This approach was considered to have a key issue in that it demonstrated a need for more precise characterisation of the meaning of the structures, as the semantics of arcs were not readily apparent (Brachman, 1983), particularly when considered with complex representations. Even in this simple example, it is fairly evident that semantic nets are unlikely to scale well. Finally, a language is required to formally define the elements of the structure.

Frame-based Ontology

Frame-based ontologies provide structured representations of objects or sets of objects (classes). In the system terminology, a frame is a named data object having a set of slots, where each slot represents a property or attribute of the object. Slots can have one or more values (called fillers); some may be pointers to other frames. They allow classes to be described as specialisations of more generic classes (Fikes and Kehler, 1985).

Gruber (Gruber, 1993) considered that five types of components: classes, relations, functions, formal axioms, and instances were required to formally specify an ontology. Two of these, classes and relations (defined in a frame by its membership and slot/attribute descriptions), will be briefly considered as an insight into the language syntax to define conceptualisations. Frames can be organised into taxonomies by superclass-of or subclass-of properties.

Class City example:

```
(define-class City (?city)
        "A city is a centre of population having a Cathedral"
:axiom-def
        (and (subclass-of city pop_collection)
        (slot-value-cardinality city city.hasCathedral 1))
:def

        (and (slot-value-type city city.hasCathedral String)
        (slot-value-cardinality city city.hasName 1)
        (slot-value-type city city.hasName String)))
```

Relation Connects example:

```
(define-relation Connects (?city1 ?city2 ?motorway)
        "A motorway connects two centres of population"
:def
        (and (component ?city1) (component ?city2) (Motorway-section ?motorway)
        (not (part-of ?city1 ?city2))
        (not (part-of ?city2 ?city1)))
```

By specifying class membership and condition-based rules, frame-based representation can support a KR system’s reasoning capability. According to Gruber, formal axioms specify statements that are always true, e.g. the following is valid: “travel from Scotland to Northern Ireland cannot be made by rail”. Therefore, any ontology specification that was inconsistent with this, e.g. rail travel between Belfast and Glasgow, could be tested using a reasoning tool, for verification of ontology consistency.

Frame-based and object-oriented approaches differ from DL in that their central modelling primitive of classes (frames), have certain attributes that apply only to the frame for which they are defined, i.e. a frame models one aspect of a domain and does not have a global scope.

Description Logics

Description Logics (DL) (Nardi and Brachman, 2002) are a family of knowledge representation languages that provide the capability to formally represent the terminological knowledge of a subject domain by expressing knowledge through a set of constructors that permit complex descriptions of concepts and roles. DL can be considered a sub-language of predicate logic.

Predicate logic (or First-Order Predicate Logic) is a KR language that, through assertions, permits both representations of complex facts about the world and, through rules of inference, derivation of new facts, i.e. on the basis that if the initial facts were true then so are the conclusions; the basic modelling primitives of predicate logic are predicates.

For example, take the geographical associations Q represents a "capital", P represents a "city", and Z represents a "country"; based on the statements "capitals are cities" and "countries must have capitals" holding true, First-Order Predicate Logic can be used to represent the statements, i.e. "capitals are cities" $(Q \wedge P)$ and "countries must have capitals" $(Z \Rightarrow Q)$. Propositional logic can then be used to prove the proposition (hypothesis) "if *capitals are cities* and *countries must have capitals* then (the conclusion is) *countries must have cities*" $(Z \Rightarrow P)$, by using the following propositional statement:

$$((Q \wedge P) \wedge (Z \Rightarrow Q)) \Rightarrow (Z \Rightarrow P)$$

In effect, the propositional statement (embodying a premise, implication and conclusion) is saying that if $((Q \wedge P) \wedge (Z \Rightarrow Q))$ then *implication* is that $(Z \Rightarrow P)$. A truth table can then be used to prove the original proposition, based on the standard *definition of implication* – see Table 1.

Table 1. Truth table to prove $((Q \wedge P) \wedge (Z \Rightarrow Q)) \Rightarrow (Z \Rightarrow P)$

Z	Q	P	Q ∧ P	Z ⇒ Q	(Q ∧ P) ∧ (Z ⇒ Q)	⇒	Z ⇒ P
t	t	t	t	t	t	**t**	t
t	t	f	f	t	f	**t**	f
t	f	t	f	f	f	**t**	t
t	f	f	f	f	f	**t**	f
f	t	t	t	t	t	**t**	t
f	t	f	f	t	f	**t**	t
f	f	t	f	t	f	**t**	t
f	f	f	f	t	f	**t**	t

The truth table provides a complete proof and shows (in the implication column headed "⇒") that when $(Q \wedge P) \wedge (Z \Rightarrow Q)$ is true, $Z \Rightarrow P$ is true. Incidentally, it may also be noted that the implication will still be true if Z is false.

This logic can then be extended in First-Order Logic assertions formed by using the propositions $Q_{(x)}$, $P_{(x)}$, and $Z_{(x)}$, in a universe of discourse where $_x$ represents the truth that “all capitals must be cities”; this can be constructed with the *universal* quantification “for all” by using operator $\forall$:

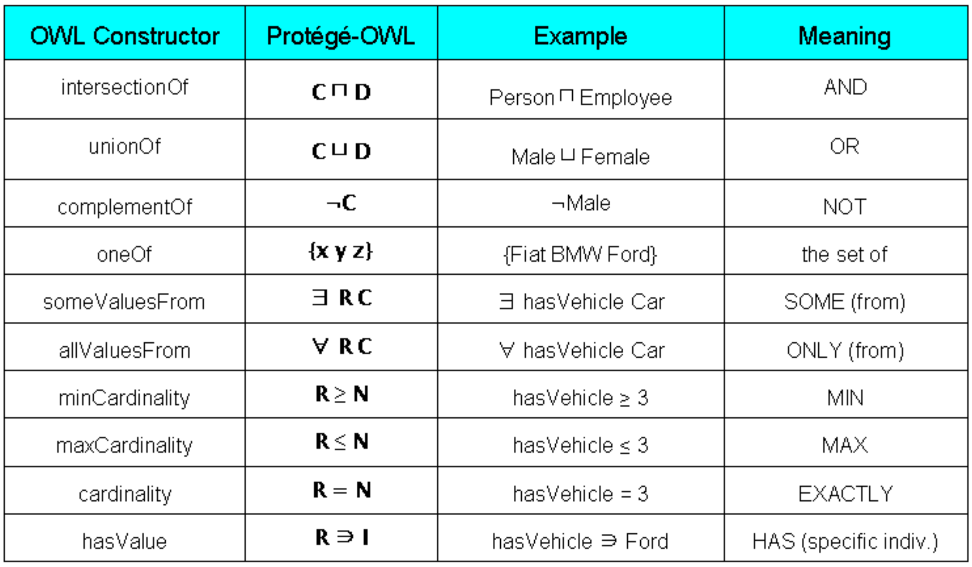
$$\forall Q_{(x)} \; (Z_{(x)} \Rightarrow P_{(x)})$$

If all capitals were not cities, the *existential* quantification, using the “there exists” or “some” operator $\exists$, could be made:

$$\exists Q_{(x)} \; (Z_{(x)} \Rightarrow P_{(x)})$$

Reasoning of DL-based concept and role restrictions can automatically derive inferred classifications and this highlights a key difference between Frame-based and DL approaches. Frame-based relies on explicit statements of class-subsumption, whereas DL is able to efficiently compute subsumption relationships between classes - on the basis of the intensional definition of these classes (Horrocks et al., 2000, Fensel et al., 2001) using asserted conditions (constructors).

DL’s use of constructors facilitates the critical issue of decidability in reasoning, to ensure consistency. The subsumption relationship can be used to express other relations between classes, e.g. transitivity, disjointness and equivalence. The ability to infer such relations is important for ontology verification and classification, particularly when the exchange, reuse and merger of ontologies constructed by different ontologists are considered. In these cases, reasoning support is vital. Examples of the range of DL constructors, syntax, and semantics are illustrated in Fig. 9, and were taken from the Protégé-OWL ontology tool.



OWL Constructor	Protégé-OWL	Example	Meaning
intersectionOf	$C \sqcap D$	Person $\sqcap$ Employee	AND
unionOf	$C \sqcup D$	Male $\sqcup$ Female	OR
complementOf	$\neg C$	¬Male	NOT
oneOf	{x y z}	{Fiat BMW Ford}	the set of
someValuesFrom	$\exists$ R C	$\exists$ hasVehicle Car	SOME (from)
allValuesFrom	$\forall$ R C	$\forall$ hasVehicle Car	ONLY (from)
minCardinality	$R \geq N$	hasVehicle ≥ 3	MIN
maxCardinality	$R \leq N$	hasVehicle ≤ 3	MAX
cardinality	$R = N$	hasVehicle $= 3$	EXACTLY
hasValue	$R \ni I$	hasVehicle $\ni$ Ford	HAS (specific indiv.)

Fig. 9. Description Logics constructors.

An equally important aspect of DLs is the distinction between TBox (*terminological* box) and ABox (*assertional* box). Concepts and roles are both described with terminological descriptions defined by constructors.

The TBox contains what is termed *intensional* knowledge and is constructed using declarations defined by a *terminology* (a controlled vocabulary); i.e. the TBox represents and facilitates the specification of subsumption axioms to describe concept hierarchies, e.g. the general properties of classes (concepts) and any relations (roles) between classes. An axiom is a proposition that is not proven but is regarded as self-evident and serves as a starting point for deducing and inferring other truths, e.g. a travel ontology might say that “all family vehicles are either cars or utilities” using a property condition $\forall \text{FamilyVehicle} \sqsupseteq (\text{Car} \sqcup \text{Utility})$, i.e. the specialisations Car and Utility are subsumed by the generalisation FamilyVehicle. Similarly, a class can be described by a constraint placed on a role, e.g. $\text{Utility} \ \forall \ \text{hasPoweredWheels AllWheels}$.

More formally, a TBox declaration of the above proposition (“all family vehicles are either cars or utilities”) could be described as corresponding to First-Order Logic, based on the following statement: “x represents the domain state that C is either a car or a utility, and D is a family vehicle.

$$(\mathrm{C} \sqsubseteq \mathrm{D}) = (\forall x)(\mathrm{C} \Rightarrow \mathrm{D})$$

An ABox contains what is termed *extensional* knowledge (or asserted knowledge) that is specific to class individuals of the ontology domain or subject context. So, equally, an ABox provides the definitions of instances (the concept and role membership assertions for instance data), including assertions or facts about the attributes of those instances; using the same controlled vocabulary used by the TBox. The ABox specifies where class instances belong, i.e. relations between classes and individuals, e.g. class $\text{Driver} \ni \text{hasValue}$ “Bob”.

Classes in DL are termed *primitive* classes, if they are specified with *necessary* conditions, i.e. if something is a member of a class then it must satisfy those conditions. However, there is no guarantee that a class will be a member of another class, e.g. Car and RailLocomotive should not be considered equivalent forms of transport just because each is described by the condition hasTraction Wheel.

Classes are termed *defined* classes, if they are specified with both *necessary and sufficient* conditions, i.e. if some other class is specified as fulfilling that condition (by a necessary condition) then it must be a member of the class having the necessary and sufficient condition. This will be seen in the next subsidiary heading, where classes Bypass and DualCarriageway, each having the condition hasFeature CentralReservation, have a subsumption relationship where Bypass is subsumed by DualCarriageway.

First-Order Logic is not concerned with such a distinction, but DL reasoners, e.g. FaCT++ (Bechhofer and Horrocks, 2000), may be required to analyse TBox and ABox statements, when validating and classifying an ontology classification by inferencing.

Description Logics in Ontology Specification and Development

Universal and existential quantifiers are some of the constructors used in Description Logics (DL) to describe domains and constrain classes and relations. Concrete representations of concepts, i.e. classes, can be specified by using logical expressions and restrictions, that use DL constructors to define class membership. For example, the concept Transport could be specified in specialisation classes RoadTransport, RailTransport, and AirTransport, where, e.g. RoadTransport might be defined using a logical expression unionOf (Car $\sqcup$ Coach $\sqcup$ Truck). A different class Commercial could be defined by specifying the membership restriction of having allValuesFrom ((Coach $\sqcup$ Truck) $\sqcap\neg$ Car).

Class descriptions are important because they provide the basis for knowledge sharing with machines, i.e. they represent explicit statements, or modelling decisions, to ensure that class individuals/instances fulfil conditions, e.g. the rules for membership of a particular class can be formally described to specify the basis on which a class can exist in a particular domain context. Further, given a skeleton class hierarchy and a set of class membership rules (asserted conditions defining possible class membership), it is possible to use a DL reasoner to classify an ontology, e.g. to (i) simply determine for each individual class, its super class or domain membership or (ii) generate a *larger* inferred hierarchy. An example of ontology classification for (i) is demonstrated in Fig. 10 a_1 and a_2, based on the following modelling statements:

Bypass $\sqsubseteq$ (Multi-laneHighway $\sqcup$ TrunkRoute) (*N*)

Bypass $\sqsubseteq$ $\exists$hasFeature.CentralReservation (*N*)

Multi-laneHighway $\sqsubseteq$ Hi-speedRoad (*N*)

TrunkRoute $\sqsubseteq$ VehicleRoute (*N*)

DualCarriageway $\sqsubseteq$ (Hi-speedRoad $\sqcup$ VehicleRoute) (*N*)

DualCarriageway $\equiv$ $\exists$hasFeature.CentralReservation (*N&S*)

The statements specify that Bypass is a sub-class of both Multi-laneHighway and TrunkRoute, which are in turn sub-classes of Hi-speedRoad and VehicleRoute respectively. Class DualCarriageway is also a sub-class of the two classes Hi-speedRoad and VehicleRoute but is a defined class, having a necessary *and* sufficient (*N&S*) asserted condition hasFeature CentralReservation, whilst Bypass is a primitive class, i.e. having only a necessary (*N*) asserted condition hasFeature CentralReservation. The class hierarchy is represented in Fig. 10 a_1.

An *N&S* condition means not only is the condition necessary for any class to be eligible for membership of the class but also that it is sufficient to determine that any individual member of a class that satisfies the condition must logically be a member of the class. A class that is defined by an *N&S* condition is known as a *defined* class.

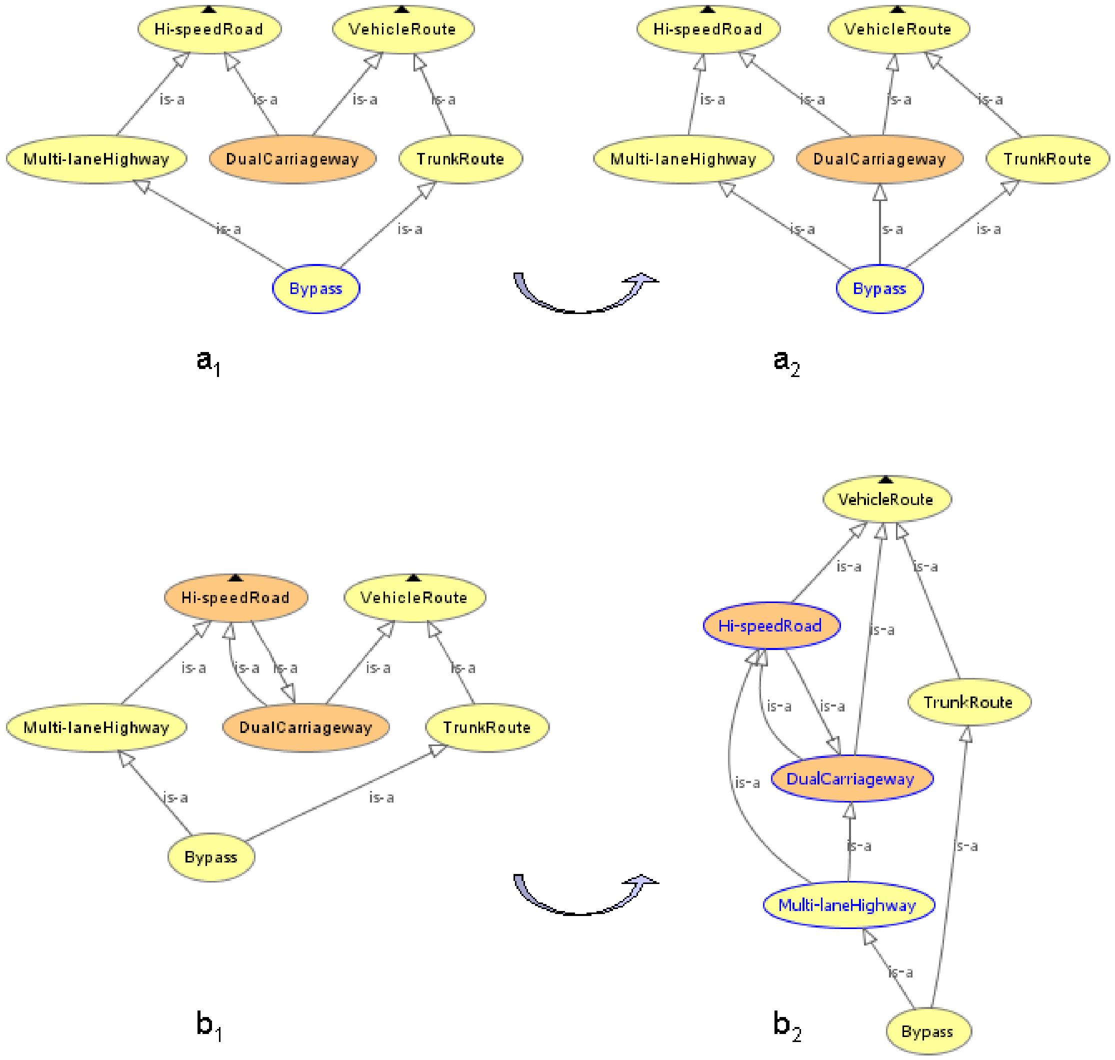


Fig. 10. Subsumption re-classification using Description Logics reasoning.

Similarly, an *N* condition means that, for any class individual to be a possible member of this class then it must fulfil the condition but the necessary condition is insufficient itself to determine that it must be a member of this class. Any class specified with only necessary conditions is known as a *primitive* class.

Based on the above asserted condition principles, and after classification using a FaCT++ DL reasoner, it was determined that as DualCarriageway is a defined class (having an *N&S* condition hasFeature CentralReservation), any class that satisfies the condition must be a sub-class of DualCarriageway. Given that Bypass has the *N* condition hasFeature CentralReservation then it must be a sub-class of DualCarriageway; therefore the ontology will show Bypass $\sqsubseteq$ DualCarriageway; this is demonstrated in the revised hierarchy in Fig. 10 a_2.

Fig. 10 b_1 and b_2 demonstrate (ii), classifying a larger inferred hierarchy, and are based on the following five statements:

Bypass ⊑ (Multi-laneHighway ⊔ TrunkRoute) (N)
Multi-laneHighway ⊑ Hi-speedRoad (N)
TrunkRoute ⊑ VehicleRoute (N)
Hi-speedRoad ≡ DualCarriageway (N&S)
DualCarriageway ⊑ VehicleRoute (N)

The statements and Fig. 10 b_1 hierarchy, appear similar to Fig. 10 a_1, i.e. having Bypass as sub-class of both Multi-laneHighway and TrunkRoute, being sub-classes of Hi-speedRoad and VehicleRoute respectively. However, the condition hasFeature CentralReservation is not applied to any classes and, whilst DualCarriageway is a sub-class of VehicleRoute, it instead has an equivalence relationship with Hi-speedRoad, making both *defined* classes, as equivalence is defined as an *N&S* condition.

Based on the above specification, the reasoner will re-classify the ontology by inferring that, Bypass must be a sub class of DualCarriageway because it is a sub class of Hi-speedRoad; which has an equivalence relationship with DualCarriageway. Therefore Multi-laneHighway ⊑ DualCarriageway applies. Finally, based on the equivalence relationship between DualCarriageway and Hi-speedRoad, and that DualCarriageway is a sub-class of VehicleRoute, the reasoner will logically infer that Hi-speedRoad ⊑ VehicleRoute applies.

Inferring Ontology Relationships between Instances and Classes

An ontology can assert relationships between instances and their class types, e.g. it may specify that class B is a sub-class of class A, (B ⊑ A), and it may also assert that instance i is a resource only of B, e.g. as typically formalised in the Ontolingua Frame-ontology (Gruber, 1992) when specifying definitions of ontology components. However, a class (or subsumption) hierarchy can make transitive assumptions, i.e. that in the above, B ⊑ A if, and only if, every instance i of B is also an instance of A; further, instances can be either an "instance-of", or "direct-instance-of" a class, depending on circumstances. However, if there were a need to identify all relevant class types for the instance i, without a reasoning tool and given the above assertions, any ontology traversal would simply determine B as the class type; further, certain applications, may require class identification somewhere between the complete list of types and the asserted base class.

The above issues can be demonstrated, on the basis that a DL reasoner has been used to determine class types based on *inferred* and *direct inferred* relationships. Fig. 11 shows three varied asserted relations for a simple ontology class hierarchy: in Fig. 11 (a), the individual city "Leeds" has been specified (asserted) to be a member of the set of instances of both class types

Regional_Centre and Government_Centre. In Ontolingua Frame-ontology, “Leeds” would be defined as simply an “instance-of” these classes.

In Fig. 11 (b), “Leeds” has been asserted as a member of the set of instances of both class types Reg_Centre and City. Each relation shows a “direct-instance-of” relationship, as “Leeds” is not also asserted to be an instance of both a sub class and its super class, i.e. Reg_Centre and Govt_Centre as in Fig. 11 (a); this is also referred to as maximally-specific and demonstrates uniqueness.

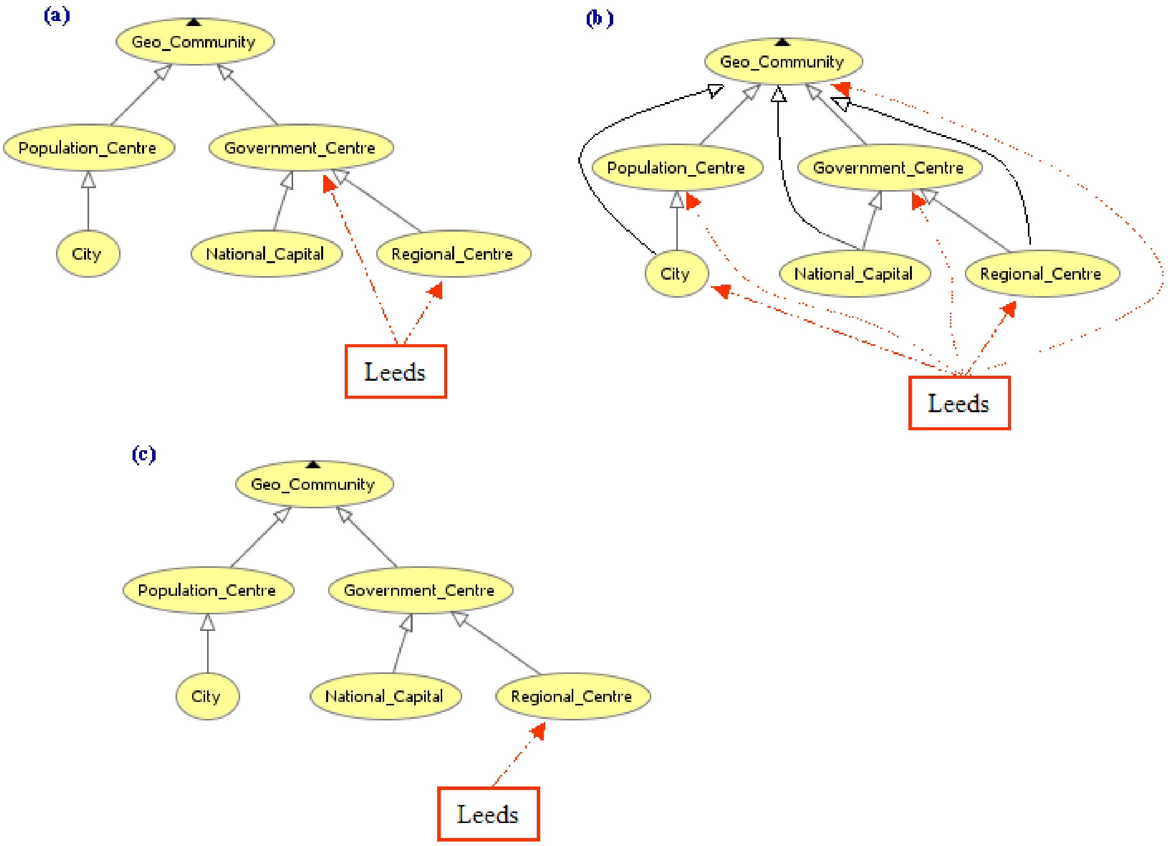


Fig. 11. Asserted, inferred and direct ontology relationships.

However, reasoning on (b) will also show all inferred relationships, firstly showing that “Leeds” has inferred instance-of relationships with class types Government_Centre, Population_Centre, and Geo_Community, and secondly, the inferred subclass-of relationships that class types City, National_Capital and Regional_Centre have with Geo_Community. Finally, Fig. 11 (a) is repeated in Fig. 11 (c) but this time is based on *direct inferred* relationship reasoning on “Leeds”; the effect, is that termed *direct graph* reasoning only shows the relationship with Reg_Centre and has “hidden” inferred relationships with Government_Centre and Geo_Community. By using inferencing techniques, a full set of relationships can be determined.

Semantic Web mark-up provides a natural application for DLs, because it will rely on ontologies to provide common terms with formalised semantics.

1.4.5 Development of Modular Ontology Concepts

A Semantic Web "lift-off" will depend on a critical mass of RDF/OWL resources being developed coupled with wider ontology acceptance and reuse - indeed, reusability is a key benefit of ontology development (Noy and Hafner, 1997). However, as ontologies can be small or monolithic, and may be designed for shared or local use, their reuse by ontology mapping could be problematic given the potential for design heterogeneity or partial overlap (redundancy) – as alluded to previously in section 1.2, regarding semantic heterogeneity.

The specification of ontology domains in this research will attempt to address issues of reuse and redundancy at the conceptualisation and modelling stage; by embracing the approach of Rector (Rector, 2003), where modularisation of ontology *concepts* represents a best practice approach motivated from the origins of database normalisation, e.g. classes are initially specified at an atomic level (i.e. self-standing). Rector advocates the "untangling" of ontologies so that primitive classes, roles and relations are decomposed (or "normalised") into a hierarchy of self-standing, disjoint trees and then subsequently combined by using relationships, definitions and restrictions to explicitly define more complex concepts.

The creation of disjoint trees helps to avoid the problems of specifying multiple inheritances. By ensuring all primitive classes exist in only one module or tree of concepts, such disjoint trees help to remove inconsistency and lack of clarity, e.g. when the hidden or inexplicit meaning can be so often hidden within complex/compound/verbose concept names of classes that really represent derivations of primitive classes, i.e. *defineables* (dependent classes). In a transport ontology, an example could be a class named Multi-levelMotorwayJunction but whilst the name would convey reasonable understanding to a human, by itself, it provides no clue to a reasoner as to what components are really described by such a concept. By creating primitives, roles and relations these can be combined to form such a defined class. By constructing complex classes in this way, a disjoint hierarchy allows a reasoner to check consistency and accurately infer subsumption (which might include multiple inheritance). Ontology development can then be considered more explicit and sound - on the basis that sound conclusions will follow from sound premises. In effect, primitives, roles and relations are used as building blocks and the following examples show how they can be used to create defined ontology concepts:

- primitives: self-standing entities, objects or forms, e.g. Structure, Process, System, Organisation;
- roles: functions e.g. CarriesRailTraffic, RailTransportRole;
- relations: concept-linking properties or binary relationships like hasRole(X,Y) e.g. TrainOperator hasRole $\exists$RailTransportRole.

The above are then combined to form definable (dependent) concepts, e.g. RailwayBridge can be defined by combining primitives, relations, and roles:

$$\text{RailwayBridge} \equiv \text{Bridge} \sqcap (\text{hasForm} \ \exists \ \text{Structure} \sqcap \text{hasRole} \ \exists \ \text{CarriesRailTraffic})$$

From the above, it should be clear that Rector offers a structured, incremental approach to meaningfully describe concepts; by modularising the semantic constructs of concepts within an ontology. This approach facilitates reusability both within and between ontologies and, indeed, the modularisation of ontology *elements* provided a lead for the method by which later examples of ontology *modules* (i.e. small ontologies defined as reusable and contextual components - in effect plug-ins to form domain and application ontologies) were developed for this research. It will also be seen later that modularity also supports the basis on which subsequent ontology-based query expansion (*OQE*) techniques have been executed because, given the use of search keywords, primitive (stand-alone/atomic) classes would seem consistent with the notion of such query terms and subsequently with *OQE* and document text matching.

Whilst a key requirement for both modularised ontology concepts and sub-domain plug-ins is that they should be self-contained and able to support reasoning in their own right (Stuckenschmidt and Klein, 2004), the modularisation of ontology elements and the capability of an ontology to serve as a component module to build a larger structure are both interlinked but different. The former represents an approach to modularise the semantic constructs of concepts within an ontology, whereas the latter represents a reusable sub-domain plug-in. Where there must be consistency is that, for module ontologies to be combined and used in conjunction with a reasoner to enable classification, they should be specified using the same semantic constructs described above.

1.5 SEMANTIC WEB ONTOLOGY LANGUAGES AND TOOLS

Based on ontology theory discussed in subsection 1.4.1, formal ontology is a key component in delivering the Semantic Web; particularly given the similarity in which both have been defined, with references to "*formal refers to the fact that the ontology should be machine-readable*" (Studer et al., 1998) and *"defined concepts"* and *"machine-readability"* (Berners-Lee et al., 2001). On this basis, the following languages and tools are relevant, i.e. those components that would either be usable in a Web environment and/or be capable of directly generating Semantic Web linked data (via RDF) and knowledge representation (via OWL).

1.5.1 Semantic Web and Ontology Languages

Human users and Web agents can view the traditional WWW as a web of document resources that are navigated by traversing hyperlinks. The effect of linking these resources is that they become "integrated" to form a global information space. Further, the Web is based primarily on HTML documents that are designed to control the visual presentation of a body of organised text and related components, e.g. describing objects such as images and interactive forms to the human interpreter. However, HTML itself has no capability to determine the semantics of what a set of characters presented could actually convey to a machine. For example, a Web page displaying the word television might also present other text, say product id: DTV-34FS, but there is no way of establishing with certainty what the text product id actually describes: it could refer to either a television, or a 34 inch screen, or a flat-screen, or it could relate to something that is not even a consumer product. HTML simply says that the span of text DTV-34FS is something that should be positioned near characters television and product id.

In response to this problem, new research initiatives were undertaken by the W3C and Tim Berners-Lee introduced "The Semantic Web" in 2001, in what has been termed a seminal article in Scientific American (Berners-Lee et al., 2001). He referred to it as *"an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in co-operation".*

Whereas the WWW represents a web of documents based mainly on HTML syntax, the Semantic Web should be viewed as a *web of data* supported by a common description framework technology that allows data to be shared and reused between organisations and by consumers. Some parts of the Semantic Web technologies have foundations based on Artificial Intelligence research, e.g. knowledge representation.

More recently, Berners-Lee launched another semantics-based initiative by outlining best practice design issues for linking diverse data sources, i.e. effectively creating a Web of *linked* data (Berners-Lee, 2006). Linked data on the Web is based on standards for representation (e.g. the Friend of a Friend (FOAF) project [1], identification, retrieval and merger, and has been adopted by an *open data* movement that is based on the W3C SWEO's Community Project Linking Open Data (SWEO, 2009); it has the goal of extending the Web by making data freely available to anyone. By publishing various *open* data sets on the Web as RDF and by setting RDF links between data items from different data sources, Web crawlers can traverse the data links and, at any time, discover new linked data published on the Web. It has been referred to as the emerging Web of Linked Data (Bizer, 2009). Therefore, whereas the Web, or *syntactic* Web, is simply about presentation to humans, the challenge of the Semantic Web is to realise the potential of the Web, by extending the capability of the Web through the use of standards, mark-up languages and related processing tools (software agents). As a result, it should be possible to execute sophisticated interpretation tasks using an extended web of machine-readable data; expressed more meaningfully by having knowledge representation coded in the page. However, Berners-Lee said the realisation of this vision would require a number of enabling stages, because Semantic Web agents and tools would require languages capable of providing far greater expressiveness than that offered in the base XML syntax (Gómez-Pérez and Corcho, 2002, Patel-Schneider and Fensel, 2002, Decker et al., 2000a).

[1] FOAF: http://www.foaf-project.org/.

The Semantic Web "tower" (Berners-Lee, 2000) illustrates a number of intermediate and related layers, comprising standards and tools – see Fig. 12. XML represents the universal syntax carrier and XML Schema the mechanism to control the structure of XML documents. A URI (Uniform Resource Identifier) provides a mechanism to define unique location references to entities and relations, whilst NS (Namespaces) enable differentiation between combinations of documents, i.e. to avoid semantic "collisions", e.g. whilst they may use similar terms/metadata/vocabularies they may refer to semantically conflicting data or domain contexts.

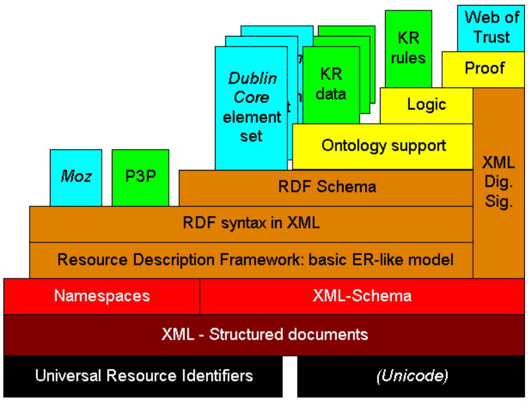


Fig. 12. The Semantic Web Tower (Berners-Lee, 2000)

The tower incorporates several key Semantic Web technologies that have been adopted as World Wide Web Consortium (W3C) recommendations, the Resource Description Framework (RDF) core language (W3C, 2004c), and the RDF Schema and OWL Web Ontology languages (W3C, 2004b). These will serve as a platform to support a standardised query language for RDF that will permit widely distributed RDF/XML data collections to support integration and function as a universal data exchange mechanism. The role of ontology in the Semantic Web is to formally describe and specify a domain context by addressing structural and semantic heterogeneities to provide a shared vocabulary that can be referenced by different applications in the subject domain. Thus, the ontology can serve as an anchor point by semantically linking information across the Web to permit heterogeneous data source integration and interoperability between contextually related domains.

New Web ontology languages have included the industry's previously de facto standards of DARPA Agent Markup Language and Ontology Interchange Language (DAML+OIL) (Hendler and McGuinness, 2000, Bechhofer et al., 2001, Fensel et al., 2001, Bechhofer et al., 2000, Connolly et al., 2001, Decker et al., 2000a), which was subsequently subsumed by W3Cs 2004 OWL recommendation (W3C, 2004b); a revision of DAML+OIL, incorporating lessons learned during its design and application. OWL will provide the enabling technology for formalised knowledge representation in ontologies.

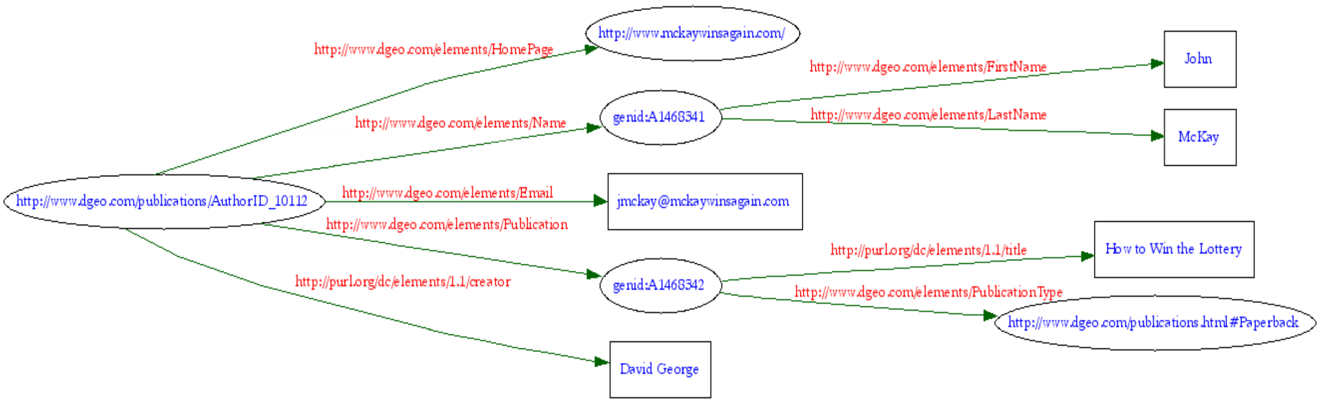
However, in recent articles called "Rethinking the Semantic Web" (McCool, 2005, McCool, 2006), it was expressed that current approaches will "*never achieve widespread public adoption*" because of it's complex format and requirement for users "*to sacrifice expressivity* *and pay enormous costs in translation and maintenance*". The evidence given for this was, somewhat paradoxically, the distinct lack of Semantic Web content currently available and the lack of a "killer application" to promote it – and yet the traditional Web surely had similar issues in its infancy? McCool also suggested that simple structures would be more practical, i.e. removing the convention of classes, relations and triples from Semantic Web formats and adding simple parameters to existing tags; with additional metadata in HTML pages to facilitate entity information exchange. However, this argument would simply ignore the fact that semantic consistency and removal of ambiguity requires ontology or vocabulary driven metadata that is both shared and agreed. Just as database interoperability suffers from structural and semantic discrepancies, without a backbone description framework, the Semantic Web would similarly suffer without order.

RDF

RDF (W3C, 2004c) is the enabling language for the Semantic Web and is a W3C recommended standard framework for describing "resources", i.e. anything that can be identified on the Web and provides the resource description framework for the Semantic Web (Decker et al., 2000b). Whilst RDF is based on XML type syntax (RDF/XML), it differs from XML, as RDF has the characteristic of being able to provide meaning through the common structures available in its data model, i.e. it provides metadata for the Web.

The underlying structure of any resource description in RDF format is a set of triples; each triple (or statement) consisting of a subject (resource object s), a predicate (subject property/attribute p), and an object (resource object or value o). The three elements (s, p and o) form a binary relationship p (s, o) or object-attribute-value: A (O,V). Alternatively, the relationship can be depicted as a labelled edge A between two nodes, O and V: [O]—A→[V].

RDF uses URI references like http://someurl, which may include a *fragment identifier* like http://someurl#*people* to identify Web resources and properties. A set of such triples is called an RDF graph and the node and directed-arc diagram in Fig. 13 provides an example; the triple forms a node-arc-node link, or directed graph. The syntax for the graph is shown in Fig. 14.



http://www.dgeo.com/elements/HomePage
http://www.mckaywinsagain.com/
http://www.dgeo.com/elements/FirstName
John
http://www.dgeo.com/elements/Name
genid:A1468341
http://www.dgeo.com/elements/LastName
McKay
http://www.dgeo.com/publications/AuthorID_10112
http://www.dgeo.com/elements/Email
jmckay@mckaywinsagain.com
http://www.dgeo.com/elements/Publication
http://purl.org/dc/elements/1.1/title
How to Win the Lottery
http://purl.org/dc/elements/1.1/creator
genid:A1468342
http://www.dgeo.com/elements/PublicationType
http://www.dgeo.com/publications.html#Paperback
David George

Fig. 13. RDF graph showing linked triples.

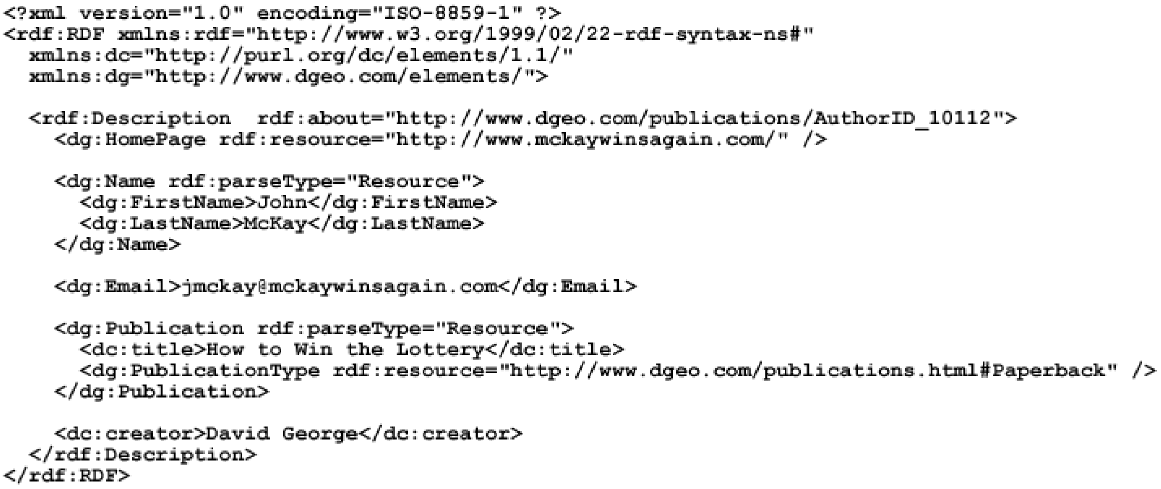
The graph example in Fig. 13 illustrates the conventions of the RDF data model:

- a subject (s) can be either an RDF URI reference, e.g. http://www.dgeo.com/publications/AuthorID_10112 or a blank node (b-node), e.g. genid:A1468341;
- a predicate (p) must also be an RDF URI reference, e.g. http://www.dgeo.com/elements/Name;
- an object (o) can be an RDF URI reference, e.g. http://www.mckaywinsagain.com/; a literal value, e.g. "David George" or again a b-node.

A b-node contains no data as such and simply serves as a parent node for a grouping of data, e.g. the b-node genid:A1468341 is the parent node for two predicates http://www.dgeo.com/elements/FirstName and http://www.dgeo.com/elements/LastName, together with their literal values "John" and "McKay" respectively.

It can be seen in the RDF example that subjects and objects are the graph nodes, with the arc direction always pointing towards the object; thus RDF/XML takes the form of a directed labelled graph, as opposed to XML that has exactly one tree representation. Further, each triple represents a *statement* of a relationship between the node elements where, in this example, the ellipse nodes represent subject/object resources, the directed edge the predicate (property), and the rectangle nodes represent literal values. RDF triples (s, p, and o) can be chained, i.e. a triple's object can in turn serve as the subject/b-node in the next triple. RDF triple chaining is particularly relevant when considering the emerging Web of *linked data* discussed earlier. An example of chained triples, in this case ((s, p, o/b-node), (s/b-node, p, o)) is shown in Fig. 13, e.g. with nodes AuthorID, Name, b-node, FirstName, "John".

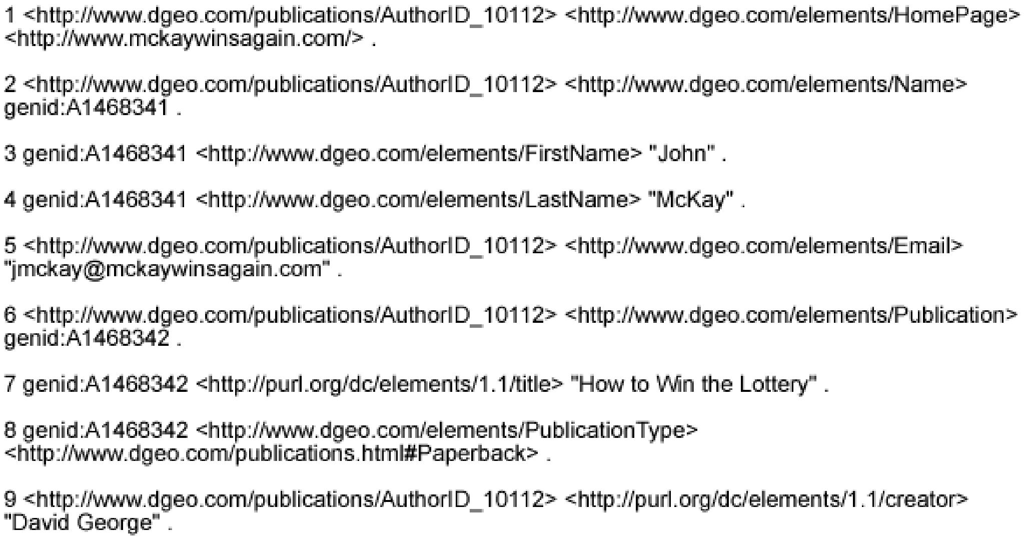
The graph also shows that subjects and predicates are identified by a URI, e.g. the subject (AuthorID) URL http://www.dgeo.com/publications/AuthorID together with the predicate (Name) URI http://www.dgeo.com/elements/Name. Alternatively, as shown in the RDF/XML syntax in Fig. 14, to minimise code reuse a short form of predicate URI has been used by binding the prefix dg: with the supporting metadata vocabulary namespace in the root RDF element, i.e. xmlns:dg="http://www.dgeo.com/elements/"; e.g. prefix dg: is used to form the tag <dg:LastName>McKay</dg:LastName>. As mentioned previously, namespace URIs and prefixes provide a means to uniquely identify a resource and thus help to differentiate similar terms that might mean different things in other resource location contexts. This can prevent ambiguity through what has been termed a "tag collision".



```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dg="http://www.dgeo.com/elements/">

  <rdf:Description  rdf:about="http://www.dgeo.com/publications/AuthorID_10112">
    <dg:HomePage rdf:resource="http://www.mckaywinsagain.com/" />

    <dg:Name rdf:parseType="Resource">
      <dg:FirstName>John</dg:FirstName>
      <dg:LastName>McKay</dg:LastName>
    </dg:Name>

    <dg:Email>jmckay@mckaywinsagain.com</dg:Email>

    <dg:Publication rdf:parseType="Resource">
      <dc:title>How to Win the Lottery</dc:title>
      <dg:PublicationType rdf:resource="http://www.dgeo.com/publications.html#Paperback" />
    </dg:Publication>

    <dc:creator>David George</dc:creator>
  </rdf:Description>
</rdf:RDF>
```

Fig. 14. RDF/XML serialisation of RDF graph.

RDF can be stored by different serialisation techniques, e.g. in the official RDF/XML syntax or by dismantling the graph into its separate triples, i.e. in N3 or N-Triple serialisation, with the latter options being better for large file storage. Fig. 14 above shows the graph in RDF/XML syntax and Fig. 15 below shows the graph separated into triples using N-Triples format. W3C's RDF validation tool [1] was used to parse the RDF/XML file and generate the RDF graph and N-Triples.



```
1 <http://www.dgeo.com/publications/AuthorID_10112> <http://www.dgeo.com/elements/HomePage>
<http://www.mckaywinsagain.com/> .

2 <http://www.dgeo.com/publications/AuthorID_10112> <http://www.dgeo.com/elements/Name>
genid:A1468341 .

3 genid:A1468341 <http://www.dgeo.com/elements/FirstName> "John" .

4 genid:A1468341 <http://www.dgeo.com/elements/LastName> "McKay" .

5 <http://www.dgeo.com/publications/AuthorID_10112> <http://www.dgeo.com/elements/Email>
"jmckay@mckaywinsagain.com" .

6 <http://www.dgeo.com/publications/AuthorID_10112> <http://www.dgeo.com/elements/Publication>
genid:A1468342 .

7 genid:A1468342 <http://purl.org/dc/elements/1.1/title> "How to Win the Lottery" .

8 genid:A1468342 <http://www.dgeo.com/elements/PublicationType>
<http://www.dgeo.com/publications.html#Paperback> .

9 <http://www.dgeo.com/publications/AuthorID_10112> <http://purl.org/dc/elements/1.1/creator>
"David George" .
```

Fig. 15. N-Triple serialisation of RDF graph.

The N-Triples format, a subset of N3, clearly demonstrates the s, p, o structure, where the elements are separated by spaces, the triple is terminated by a period '.' and URIs are enclosed by angle brackets.

[1] RDF validation tool: http//www.w3.org/RDF/Validator/.

RDF Schema

Whereas the role of RDF is to provide a basic object-attribute-value data model for metadata, RDF Schema (W3C, 2004d) is a vocabulary, or ontology modelling language, based on RDF; in effect, it comprises a set of rules to define the key components of a domain and how they relate to each other. This vocabulary enables an ontologist to define classes, properties and class hierarchies, and specify property domain and range restrictions; this provides semantics for subsumption hierarchies.

Incrementally, an Ontology layer adds more vocabulary for describing properties and classes, e.g. disjoints (relations between classes), cardinality, and equivalence. Using RDF Schema, it would be known which classes existed and what their properties were, whereas with an ontology layer it would also be possible to say when two classes are the same or whether properties have multiple values. Semantic Tower layering is shown in the following examples:

- RDF enables assertion of facts e.g. Person "45342" is named "John".
- RDF Schema facilitates vocabulary description to describe things, e.g. Person "45342" is a subClassOf LivingPerson.
- Ontology describes relationships between vocabularies and classes, e.g. "the set of Person in ontology O^1 are the same as the set of User in ontology O^2".

A further distinction between RDF and RDF Schema can be demonstrated in the graph in Fig. 16, available from an early W3C RDF Schema Working Draft (W3C, 2002).

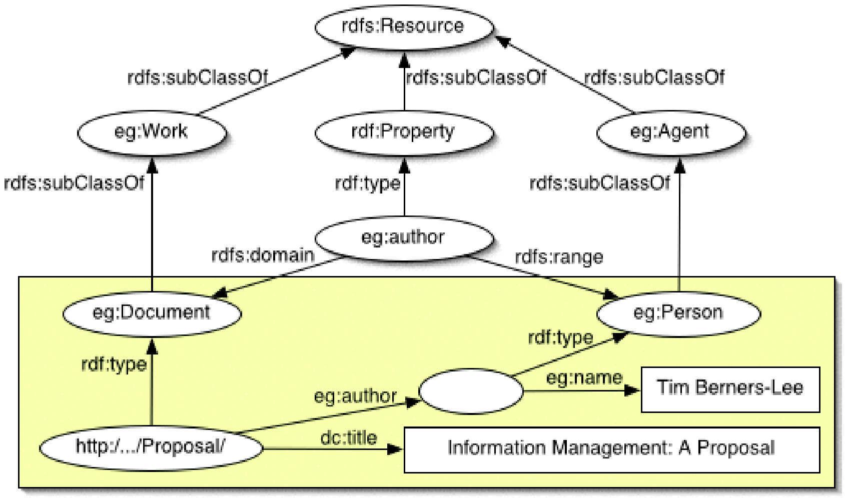


Fig. 16. Relation between RDF Schema and RDF data (W3C, 2002).

The W3C graph in effect represents two layers: the first illustrates how the RDF vocabulary rules can be used to describe real world objects, in terms of their class membership and properties that are used to relate class members, and interfaces with the second layer (shaded area) to demonstrate the link to RDF application-level data.

RDF Schema comprises a set of classes, e.g. Resource, Class, and Property. The class rdfs:Resource represents everything described in RDF and all real world things are either members (individual sub classes) of rdfs:Resource (e.g. rdfs:Class and rdfs:Property) or instances of class members. The concept rdfs:Class represents a generic type or category of Resource, whereas rdf:Property represents those resources that are RDF properties, i.e. RDF properties are sub classes of rdf:Property, e.g. rdf:Type and rdfs:subClassOf denote membership of a class, rdfs:Domain represents a domain class of a property and rdfs:Range represents a range class of a property.

The Fig. 16 example demonstrates some identifiable triples:

eg:Document rdfs:subClassOf eg:Work (a document is a work)
eg:author rdf:type rdf:Property (author is a type of property)
eg:Person rdfs:subClassOf eg:Agent (a person is an agent)

Further, the eg:author property relates an eg:Document to an eg:Person, i.e. in the domain of *document* the property *author* has a value range *person*. Finally, it can be interpreted that the *proposal* is *titled* “Information Management: A Proposal” whose *author* is a *person* named “Tim Berners Lee”. Clearly, RDF Schema enables combinations of classes, properties and values to be used together in a meaningful way.

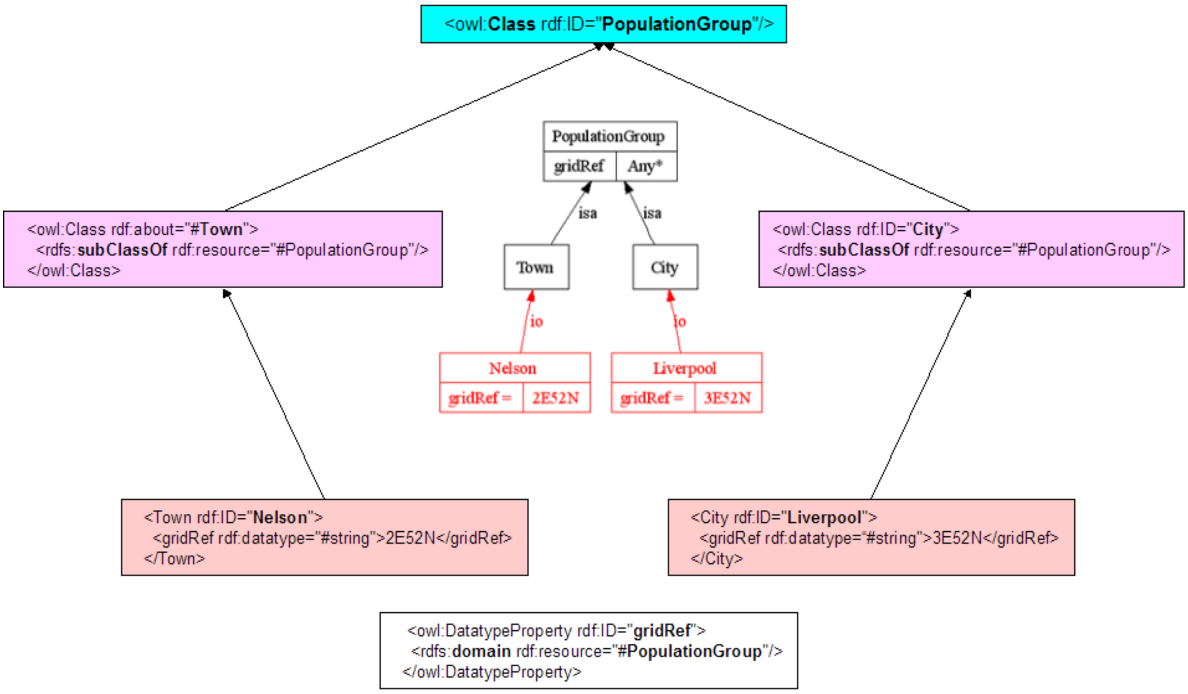
OWL

An OWL ontology is basically an RDF graph. There have previously been various ontology representation languages for the Semantic Web (Gómez-Pérez and Corcho, 2002); the latest is the OWL Web language, which is built upon RDF Schema and derived from the DAML+OIL logic-based ontology language (Horrocks, 2002). OWL is the W3C recommendation for Ontology representation in the semantic Web (W3C, 2004b). It has three species or sub-languages that demonstrate increasing expressiveness beyond the basic provisions of RDF Schema, i.e. OWL Lite, OWL DL and OWL Full.

- OWL Lite is syntactically the simplest sub-language and is suitable for basic class hierarchy and constraints, e.g. it is useful for translation from thesauri and simple taxonomical classifications.
- OWL DL is more expressive and is based on the Description Logic paradigm (a decidable fragment of First-Order Logic, i.e. decidable in finite time) and supports automated DL reasoning, e.g. subsumption reasoning to compute classification hierarchies.
- OWL Full provides the highest level of expression and is suitable where expression is more important than decidability. With OWL Full it is not possible to compute with automated reasoning tools.

The three OWL variants contain sequences of *classes*, *properties*, *relationships* and facts. OWL DL and Full allow explication of formal axioms, i.e. the specification of constraints or self-evident truths that are accepted as the basis of reasoning; axioms are used to constrain the meaning of concepts, verify ontology consistency and infer classification hierarchies.

Like RDF classes, OWL classes are the concrete representation of concepts and are associated with a set of *individuals* (the class extension). The individuals in the class extension are termed the *instances* of the class. Whereas instances represent objects in the domain of discourse, *properties* are binary *relationships* on classes and their instances, i.e. the relation between two classes or individuals. There are some important considerations to remember in OWL. OWL does not use a unique name assumption (UNA), in that two different names can refer to the same instance, e.g. "Bard" and "William_Shakespeare" can each refer to the same person. Equally, the designer must, for example, explicitly say "C1 sameAs C2", or else they might be the same or might be different. Fig. 17 shows a subsumption hierarchy fragment of a simple geographical concept defined using OWL syntax.



```
<owl:Class rdf:ID="PopulationGroup"/>
```

```
<owl:Class rdf:about="#Town">
  <rdfs:subClassOf rdf:resource="#PopulationGroup"/>
</owl:Class>
```

```
<owl:Class rdf:ID="City">
  <rdfs:subClassOf rdf:resource="#PopulationGroup"/>
</owl:Class>
```

```
<Town rdf:ID="Nelson">
  <gridRef rdf:datatype="#string">2E52N</gridRef>
</Town>
```

```
<City rdf:ID="Liverpool">
  <gridRef rdf:datatype="#string">3E52N</gridRef>
</City>
```

```
<owl:DatatypeProperty rdf:ID="gridRef">
  <rdfs:domain rdf:resource="#PopulationGroup"/>
</owl:DatatypeProperty>
```

Fig. 17. OWL representation of subsumption hierarchy.

In the context of Fig. 17's domain, it might be the convention that membership of the class City is constrained such that all individuals of City must have a cathedral. In OWL, classes are defined by descriptions (asserted conditions) that specify the conditions for class membership, i.e. City ∀ hasFeature Cathedral. Sub-classes specialise (i.e. are subsumed by) their super-class and are defined by an "is-a" relationship; this is specified in OWL using syntax owl:SubClassOf (in DAML+OIL it is rdfs:SubClassOf). In this example, instances are identified in red, e.g. rdf:ID="Liverpool".

Fig. 16 demonstrated the relationship between RDF and RDF Schema layers; Fig. 18 extends this by showing the relationship between the RDF Schema and OWL layers.

Whilst OWL builds on RDF and RDF Schema, in terms of expressiveness, OWL components have a sub class relationship with RDF Schema, e.g. the OWL layer in Fig. 18 shows that owl:Class and owl:ObjectProperty are sub classes of rdfs:Class and rdf:Property, so that the instance layer shows the assertions of the "Manchester" and "M62" as instances of classes owl:City and owl:Motorway respectively, and their respective super classes owl:PopGroup and owl:Highway are class types rdfs:Class.

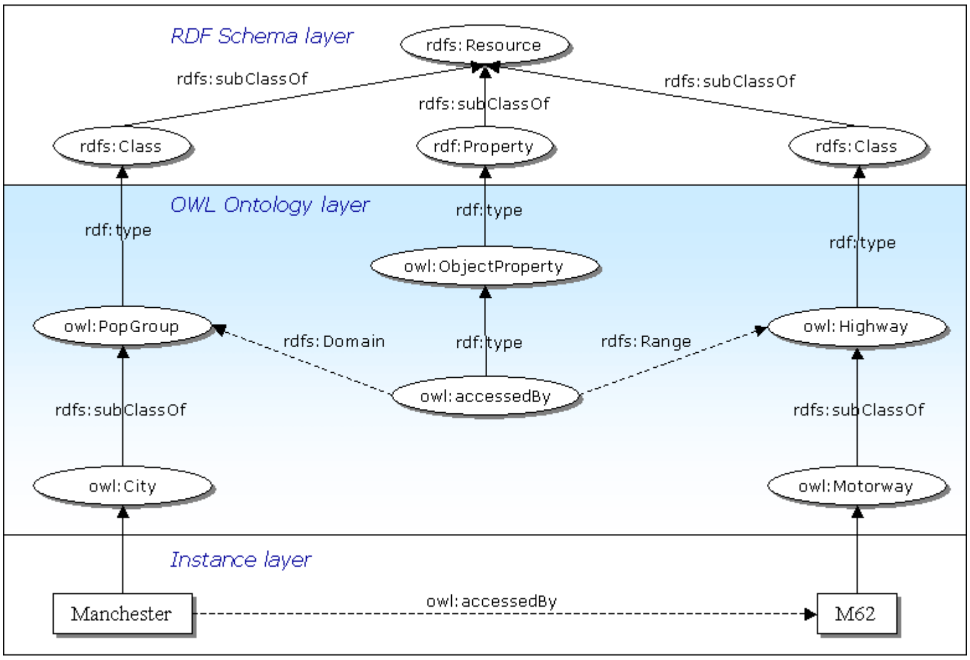


Fig. 18. Graph of relations between RDF Schema and OWL Layers.

The object property owl:accessedBy is a property type owl:ObjectProperty, with the property relating to the domain owl:PopGroup and having range values of class type owl:Highway; this is reflected in the instance layer where the ontology describes "Manchester" as being accessedBy the "M62".

1.5.2 Semantic Web Tools

There are a multiplicity of tools, both commercial and open-source, that have been the subject of various surveys (OntoWeb, 2002, Gilbert and Butler, 2003). Many Ontology and Knowledge Base tools are Java-based, given the development of platform independent applications and applets for deployment over the Web, and this was a deciding factor in tool selection.

The Java language will have an influential role in the future Semantic Web, given that JADE (Java Agent DEvelopment Framework) is a software framework to facilitate the development of Multi-Agent Systems (MAS), in compliance with the FIPA [1] specifications for interoperable intelligent multi-agent systems (Bellifemine et al., 1999). FIPA compliant commercial and open-source software agent tools are written in Java.

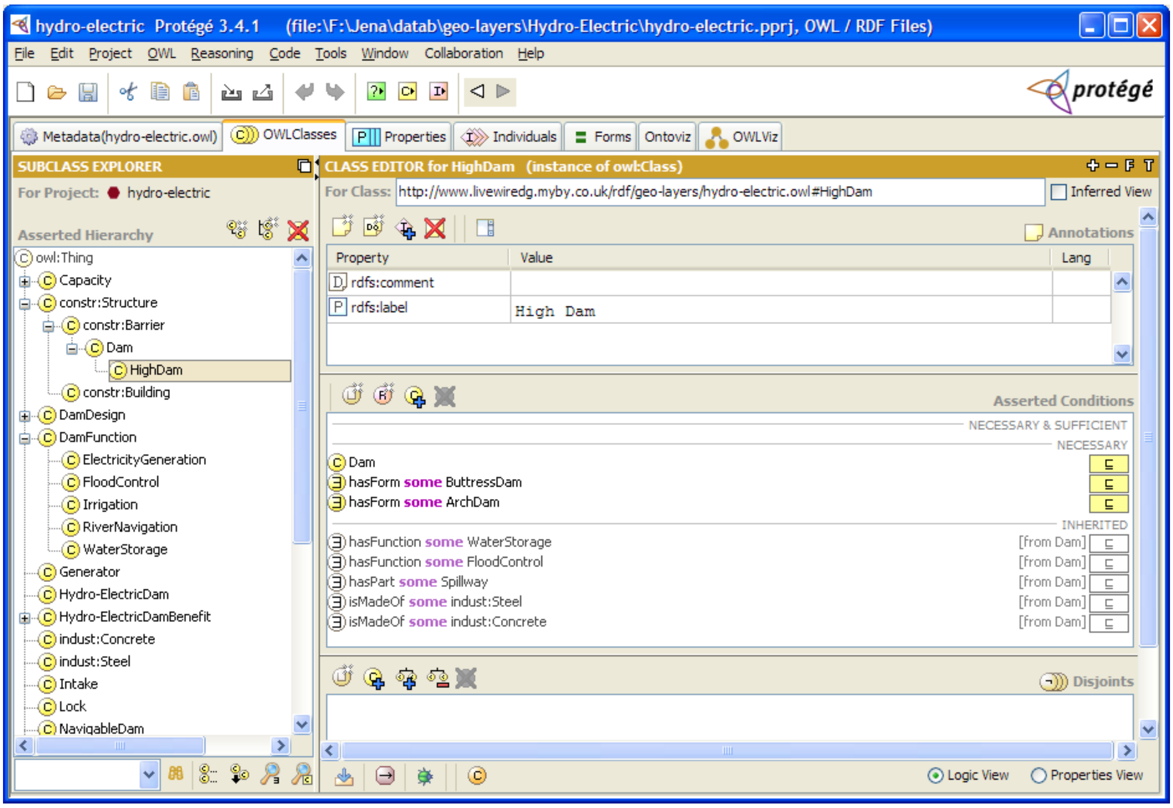
Several prominent tools have been identified because of their interrelationships between each other and their ability to programmatically generate or store Semantic Web content. Ontologies are invariably authored using an ontology editor and the widely used Protégé was selected for the ontologies developed for this research; these ontologies were components in the semantic search query expansion experiments, using a prototype search tool termed SemSeT (Semantic Search Tool), which was developed using the Jena API toolkit. Both Protégé and Jena are summarised next.

Protégé

Protégé is an ontology editor evolved out of various artificial intelligence (AI) and knowledge-modelling projects conducted at the Medical Informatics group at Stanford University (Noy et al., 2001). Protégé is a Java-based, free open-source knowledge modelling application to construct conceptual models and knowledge bases, in an application or platform independent way, as models can be developed using Protégé-Frames or Protégé-OWL editors and be saved in various formats, e.g. XML, UML and RDF/XML, using storage plug-ins.

The tool (see Fig. 19) supports numerous plug-ins (Knublauch, 2003) for knowledge model visualisation tools and reasoners, including OntoViz, OWLViz, Algernon, RACER, FaCT++ (Bechhofer and Horrocks, 2000). The gradual development of Protégé and its component plug-ins represents the collective effort of a number of research groups including Manchester University/CO-ODE group. In particular, Protégé uses DIG compliant reasoners (Description Logic Implementers Group) to compute subsumption relationships between classes and detect inconsistent classes, i.e. given an initial hierarchy plus a set of membership rules, the reasoner's job is to generate any *larger* inferred hierarchy.

[1] FIPA: IEEE Foundation for Intelligent Physical Agents – http://www.fipa.org/.



hydro-electric Protégé 3.4.1 (file:\F:\Jena\datab\geo-layers\Hydro-Electric\hydro-electric.pprj, OWL / RDF Files)

For Project: hydro-electric

- owl:Thing
  - Capacity
  - constr:Structure
    - constr:Barrier
      - Dam
        - HighDam
    - constr:Building
  - DamDesign
  - DamFunction
    - ElectricityGeneration
    - FloodControl
    - Irrigation
    - RiverNavigation
    - WaterStorage
  - Generator
  - Hydro-ElectricDam
  - Hydro-ElectricDamBenefit
  - indust:Concrete
  - indust:Steel
  - Intake
  - Lock
  - NavigableDam

CLASS EDITOR for HighDam (instance of owl:Class)

For Class: http://www.livewiredg.myby.co.uk/rdf/geo-layers/hydro-electric.owl#HighDam

Property	Value	Lang
rdfs:comment		
rdfs:label	High Dam	

NECESSARY

- Dam
- hasForm some ButtressDam
- hasForm some ArchDam

INHERITED

- hasFunction some WaterStorage [from Dam]
- hasFunction some FloodControl [from Dam]
- hasPart some Spillway [from Dam]
- isMadeOf some indust:Steel [from Dam]
- isMadeOf some indust:Concrete [from Dam]

Fig. 19. Class specification using the Protégé Ontology editor.

OWLViz allows asserted and inferred classification hierarchies to be visualised. The OWL-plugin allows model processing using the OWL ontology language (Horridge et al., 2004). Other features, amongst many, include UML, XMI, and Prolog plug-ins and support for Import/export of Protégé ontologies from/to Jena-based persistent storage.

Jena Semantic Web Framework for Java

The Jena API Framework (McBride, 2002) is an open source Semantic Web Java programming toolkit that implements RDF and OWL Semantic Web language recommendations to allow RDF-based files to be parsed and components to be abstracted. The toolkit has been used extensively during this research, initially to develop a number of Semantic Web interfaces for RDF and Ontology file interrogation/manipulation trials and then, more importantly, to develop the research search tool for the proposed experimentation. Currently in version 2.6.2, the Jena Semantic Web Framework was initially developed in the HP Labs Semantic Web Programme (HP-Labs, 2005) and supports ontology concept description, ontology management, concept manipulation, data integration and query. Jena uses packages that provide Java libraries for a developer use in a programmatic environment interacting with RDF, RDF Schema, DAML+OIL, and OWL technologies.

The RDF API provides methods for manipulating and querying an RDF model as a set of RDF triples; together with writers and parsers for RDF/XML and N-Triples. Jena also provides for persistent storage of RDF and OWL models, provides persistent storage of RDF data in relational databases engines and includes support for JDBC drivers for MySQL, Oracle, PostgreSQL databases. The Jena2 Ontology API is language neutral, supports OWL, DAML+OIL and RDF Schema ontologies, and provides various iteration methods for traversing and extracting classes and instances; it also includes a document manager process for managing an imported ontology.

Jena has a reasoner subsystem with configured OntModel rule sets for RDF Schema and OWL. These ontology reasoning rule sets can be used to construct *inference models*, which show the RDF statements entailed by the data being reasoned over, i.e. by deriving additional truth statements from one statement. Jena also provides support for a number of reasoners via its inference API. In addition, RDF-based query engine support is based on SPARQL (W3C, 2005), a W3C recommendation "data-oriented" query language, i.e. it only queries the information held in database models because the language has no inference capability.

1.6 INFORMATION RETRIEVAL BY SEARCH ENGINE

The World Wide Web is an inexorably expanding global information space. The growth in data resources, data form, contextual mix and multi-lingual content means that Web search engines can only provide the most loose form of answer set integration, in the form of a ranked list of potentially relevant documents to the user. The three main Web search engines, Google, Yahoo and MSN Search, account for about 85% of searches and employ what might be termed traditional search methods; the question is, can the Semantic Web be used to *embed* semantics-based search in traditional IR, to exploit the mass of non-RDF based data, or will the solution lie in Semantic Web search being applied to RDF data sources?

1.6.1 Traditional Search

Early search was based on Boolean search, where words and phrases can be combined using Boolean operators, e.g. AND (+), OR, NOT (-), to restrict, expand, or define a search; document ranking was not critical in a Boolean system (Singhal, 2001). However, traditional IR (SIGIR, 2008, TREC, 2008) predominantly focuses on keyword-based methods, enhanced by statistical query expansion, to generate a ranked (weighted) list of potentially relevant documents optimised in terms of search effectiveness - typically precision and recall (P&R) (van Rijsbergen, 1979). Similarly, commercial Web search engines rely on matching query terms with indexed documents and use query expansion to improve the effectiveness of search results, e.g. adding synonyms to increase recall and also precision; they also analyse link relations in hypertext documents, e.g. PageRank link analysis (Google, 2008). Research has also been conducted on deducing the context of a document collection, e.g. by using stemming, clustering/term co-occurrence techniques. Stemming involves linguistic analysis to identify the root element of a word and then returning all documents containing the root; a similar approach is to generate a set of variations of the term, by appending and removing prefixes and suffixes to a query term as appropriate, and then also using those terms in an expanded query. Google makes use of synonyms and stemming in search. However, expanding a query with stem sets and synonyms may increase recall but it can adversely affect precision.

IR uses various algorithms to determine document content relevance. Two prominent models are the vector space model (VSM) (Salton et al., 1975) and probabilistic model (Sparck Jones et al., 2000). In the VSM a document is represented as a vector space (a variable quantity that is an aggregate of components) by recognising the existence and non-existence of query terms in the text in vectors that return values either greater than zero or at zero so that a collection of vectors can be added together and the values modified by relevance weightings to determine the relevance score of a document. The probabilistic model is based on ranking documents by estimating the probability of relevance, where relatedness or similarity are calculated by probabilistic inference that co-occurrences of terms (term clustering) will be distributed differently in query relevant and non-relevant documents. Term clustering is founded on the Association Hypothesis (van Rijsbergen, 1979), i.e. that a set of related terms (e.g. lexical semantic relations) in a document collection would co-occur within documents in the collection. By exploiting this hypothesis, co-occurring terms can be clustered and expansion terms then selected from those clusters containing the query terms; thus, the probabilistic model has been important in query expansion approaches.

However, the above search approaches are unlikely to return a page containing none of the original query terms, even if it had semantically related ones. Therefore, rather than relying solely on IR data synthesis approaches, could Semantic Web ontology representation languages help Web users retrieve relevant information sources more effectively, by enabling search tools to increase the weighting of documents that have other terms that are query related? If those other terms match query relevant ontology concepts semantically related to the original query terms, they could then be used in query expansion. For this initial research in comparing the use of keywords against *OQE*, the VSM relevance measure was selected.

1.6.2 Query Term Weighting

The VSM has been extended by a classic measure for term-weighting using the *tf-idf* algorithm (Spärck Jones, 2004), i.e. for term frequency (*tf*) against inverse document frequency (*idf*), to give a weighted statistical measure of how important a term is to a document in a document corpus (*tf-idf*). By using this approach, a term's importance is increased by its frequency in the *document* (*tf*) but is reduced by the frequency of the term in the corpus (*idf*).

To achieve document relevance ranking, the *tf-idf* measure calculates the sum of a document's term weights: where *tf* represents the frequency *F* of any term *t* in document *d* (i.e. F_{td}) and *idf* is the inverse document frequency calculated by the *log* of the total number of documents *D* in a corpus divided by the number of documents n containing term t. A *term* weight vector W_{td}, e.g. for term t_i is then expressed as:

$$W_{td} = \sum_{t_i \in d,\, d \in D} F_{t_i d} * \ln\left(\frac{D}{n_{t_i}}\right)$$

To minimise the generation of exaggerated weightings, i.e. when documents contain excessively repeated terms, the frequency F_{td} for each term can be normalised by dividing it by the highest term frequency maxF_{td} found in the document:

$$W_{td} = \sum_{t_i \in d,\, d \in D} \left(\frac{F_{t_i d}}{\max F_{td}}\right) * \ln\left(\frac{D}{n_{t_i}}\right)$$

In turn, term weights determine the weight vector for a document d (W_d), i.e. a document weight vector representing multiple matched terms is the sum of all matched term weights:

$$W_d = \sum_{t_i,,,\, t_n \in d} [\, W_{t_i d},\, ,\, , W_{t_n d}\,]$$

The resulting combined *tf-idf* value can be used in P&R measures to determine search effectiveness.

1.6.3 Search Effectiveness: Precision and Recall

The IR community has traditionally evaluated search effectiveness by measuring the P&R achieved in a search process (van Rijsbergen, 1979); where P&R are defined in the following set-based measures:

$$P = \frac{|\ relevant\ documents\ \cap\ documents\ retrieved\ |}{|\ documents\ retrieved\ |}$$

$$R = \frac{|\ relevant\ documents\ \cap\ documents\ retrieved\ |}{|\ relevant\, documents\,|}$$

However, P&R do not generate ranked order and are often contradictory in that improvement in one can adversely affect the other. Given, that search engines often return thousands of hits and users are unlikely to view more than the first few result pages, precision is most important.

A determination of search effectiveness in identifying relevant documents can be achieved by applying the *tf-idf* algorithm results in a graph of precision against recall, by plotting the cumulative returned document precision values say for every 10% interval of recall – as will be seen later. However, traditional search methods tend not to return potentially relevant documents that contain *none* of the terms entered in a user's query and therefore this limitation has to be considered with existing search engine P&R results.

1.6.4 Semantic Web and Search

Based on a review of near-term prospects for the Semantic Web (Benjamins et al., 2008), and an examination of commercial search engines, there appears to be little evidence that the search engines are providing ontology-based search methods. An examination of use cases (W3C, 2008) also fails to show that this challenge is being fully exploited, either by Semantic Web communities or commercial search engines. And yet, exploitation of the expressivity of the ontological specification of concepts and relations may offer a valuable benefit in improving recall of relevant documents; by query expansion techniques that give added weight to those documents containing wider, contextually relevant text, i.e. terms that can be validated or found in a query context-relevant ontology. The question might be, however, would the size of an ontology, e.g. in terms of generalisation, specialisation, and application/domain coverage, adversely affect search processing overhead and ontology context management?

Ontologies have featured in various academic search initiatives:

i. crawler-based locators of RDF and ontology resources, e.g. Swoogle (Ding et al., 2004) and Sindice (Oren et al., 2008); search support in specialist knowledge domains, e.g. bioinformatics and the Gene Ontology (Stevens et al., 2000, Ashburner et al., 2000);

ii. international organisation support, e.g. World Bank and Organisation for Economic Co-operation and Development (OECD) (Kim, 2005) and in legal document search (Berrueta et al., 2006), where ontology query uses technical terms to find related information, terms and documents;

iii. other research involving word synsets, sense definition-based expansions, and *OQE*, include: a review of *OQE* success factors (Bhogal et al., 2007); exploitation of ontological relations (Lei et al., 2006, Fang et al., 2005); word sense disambiguation in semantic network-based sense definitions (Navigli and Velardi, 2003); “hybrid” search combining ontology and keyword based IR results (Bhagdev et al., 2008) and earlier work on lexical-semantic query expansion work (Voorhees, 1994). *OQE* often uses reasoning-based semantic query languages to extract query expansion concepts.

Commercial semantic search has included natural language processing search companies Hakia (Hakia, 2008) and Powerset (Powerset, 2008), where both use ontologies to support general document/text search.

As mentioned in subsection 1.5.1, a further development has gained increasing prominence during the last 2/3 years, i.e. RDF-based browsing research has been focused on the emerging Web of Linked Data; an additional layer interfacing with the traditional document Web, where links connect self-describing RDF files (i.e. an application can resolve unfamiliar vocabulary by identifying definitions of vocabulary terms). As anyone can publish and link data to the Linked Data Web, and the data is published on the basis that it is open data, new data sources can therefore be identified at application runtime; indeed, more and more data providers are adopting the Linked Data principles. This is resulting in a rapidly expanding data space, referred to as a "data cloud", embracing topics ranging from media to geographic data, life sciences, publications and social/user-generated content. The current extent of this can be appreciated in the data cloud (SWEO, 2009) shown in Fig. 20.

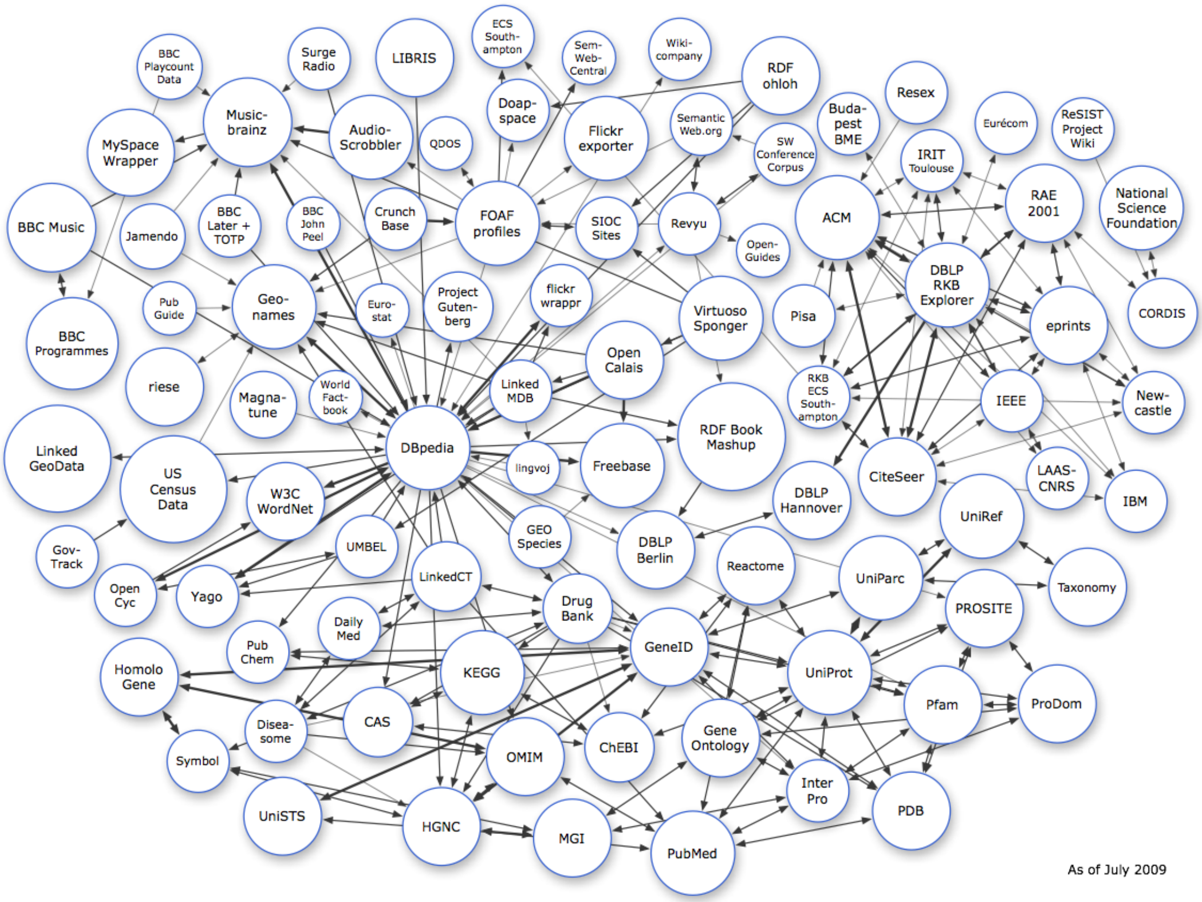


Fig. 20. The current extent of Data Cloud of Linked Data.

A number of application initiatives are in progress to navigate the Web of Linked Data, e.g. data browsers are being developed like Tabulator [1], Disco [2] and Marbles [3]. Alternatively, the data can be crawled and extracted using existing semantics-based search engines, e.g. Swoogle (Ding et al., 2004), Sindice (Oren et al., 2008). Some search engines are supporting open Web standards for describing connections between people, i.e. the social infrastructure of the Web, and offer methods to retrieve such data, e.g. Google's Social Graph API [4] indexes the public Web for XHTML Friends Network (XFN) [5] and FOAF.

[1] Tabulator: http://www.w3.org/2005/ajar/tab.

[2] Disco: http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/.

[3] Marbles: http://marbles.sourceforge.net/.

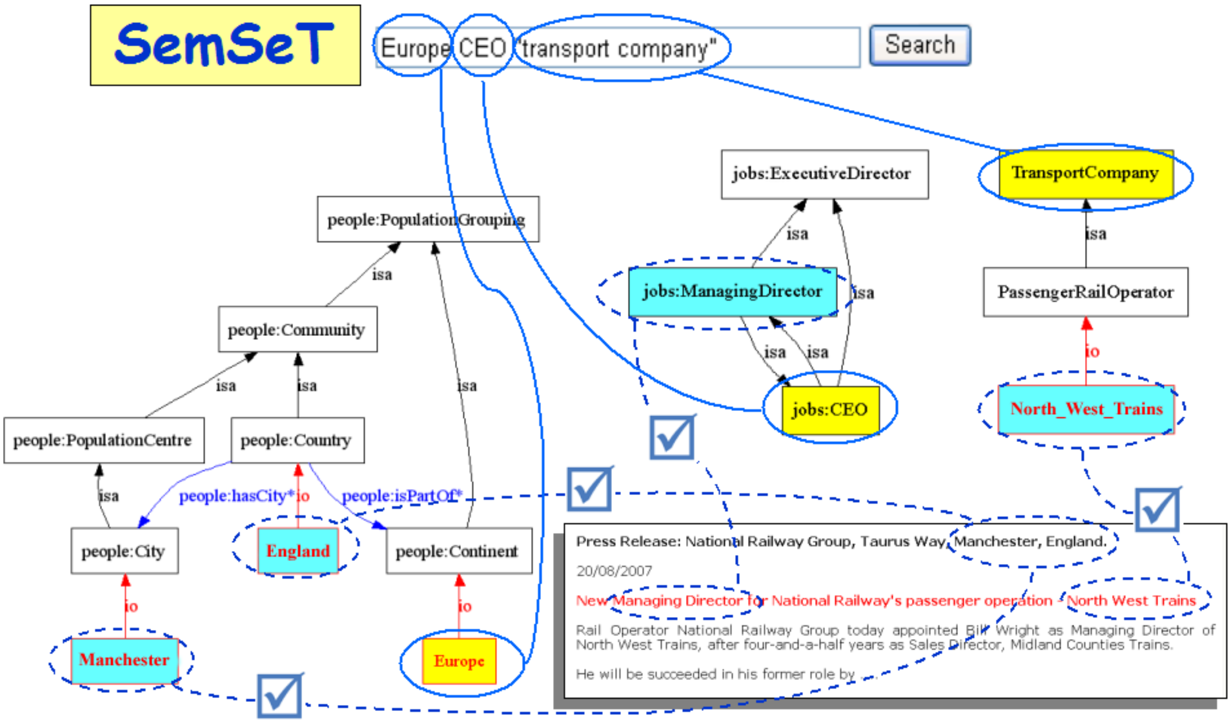
[4] Google's Social Graph API: http://code.google.com/apis/socialgraph/.

[5] XFN: http://gmpg.org/xfn/.

Regardless of the initiatives underway above, how might a semantic search tool be more effective than traditional search methods and what impact might such a tool have in improving precision and recall? Assuming that search will likely involve interaction with a contextually relevant vocabulary, some important considerations emerge.

- How would the search process relate search terms to a vocabulary/ontology and how would an ontology hierarchy be traversed?
- Would the richness of axioms describing a domain have an impact, e.g. using asserted conditions to defining class membership and relation classes? See subsections 1.4.4, "Description Logics in Ontology Specification and Development" and 3.2.5.

A semantic search tool might enhance P&R performance measures, in that *OQE* could improve recall by returning relevant Web pages containing none of the user's original query terms, e.g. if search query terms "Europe", "CEO" and "transport company" can only be matched in a query-relevant ontology context, SemSeT could search for semantically related concepts by traversing the ontology class hierarchy, i.e. from TransportCompany to find *North_West_Trains*, from CEO to find Managing Director, and from *Europe* via classes and relations to find *Manchester* and *England* - and then search for those terms within documents as depicted in Fig. 21.



Press Release: National Railway Group, Taurus Way, Manchester, England.

20/08/2007

New Managing Director for National Railway's passenger operation - North West Trains

Rail Operator National Railway Group today appointed Bill Wright as Managing Director of North West Trains, after four-and-a-half years as Sales Director, Midland Counties Trains.

He will be succeeded in his former role by ....

Fig. 21. A non-keyword matching document hit using Semantic Search.

However, as a user is unlikely to examine more than the first few pages of a search engine list of potentially relevant documents, a key question (beyond improving recall) must be: what impact would *OQE* have on search precision? Finally, the semantic correlation between ontology concepts during the ontology traversal process will have a bearing on the degree of relevance that applies to any such concept relative to a base (original) query term in any *OQE*. This issue is considered later in subsection 3.2.5.

1.6.5 Ontology-based Query Expansion

As mentioned in subsection 1.6.1, the major IR conferences tend to focus on traditional keyword-based search, supported by statistical query expansion, as opposed to formal semantics-based query expansion, despite the existence of research on sense-based query expansion and *OQE* over the past 10-15 years. Query expansion seeks to overcome potential ambiguities in natural language and the challenge of using single terms to represent and locate relevant information sources (Bhogal et al., 2007).

Early query expansion work involved exploited lexical semantic relations, or synsets, using WordNet. In 1994, experiments were conducted using manually expanded and disambiguated queries over a TREC document collection, using synonyms and other semantic relations from WordNet (Voorhees, 1994). Voorhees found that query expansion driven from full TREC query topic statements produced minimal effect on P&R, whereas expansion was beneficial on smaller, summary TREC query statements. Voorhees also found that assigning lower weights to added concepts enhances retrieval accuracy; this outcome was of particular interest because it will be seen later that some SemSeT experiments, using higher weights, produced improved results. Related work (Gonzalo et al., 1998) identified marked improvements in relevant document retrieval, by expanding a query using indexed WordNet synsets, however, this required the test collection (both queries and documents) to be disambiguated to make it work effectively; without (manual) disambiguation, the synset indexing approach was only as good as standard word indexing - at best.

In an examination of *OQE* using semantic networks (Navigli and Velardi, 2003), queries on a TREC document collection were expanded with terms found in WordNet synset sense definitions (*glosses* based on the semantic domain) and the results were compared to using taxonomic (hierarchical) relations in a sense-based query expansion. The conclusions were, firstly, that other ontology derived semantic relations (expansions based on the words in *glosses*) were more search effective than sense-based query expansion (e.g. synonyms, hypernyms and hyponyms) and, secondly, that expanding a query with terms that on a probabilistic basis frequently co-occur with query terms, because they belong or relate to the same semantic domain (e.g. aircraft and pilot/airport), is better than Semantic Web sense-based query expansion work, which it was felt had not produced strong evidence for its effectiveness. Word sense disambiguation was cited as a key problem with sense based query expansion and that high precision was more important in query expansion than recall. Interestingly, query expansions conducted using expanded glosses, i.e. including the synsets of gloss words, were less favourable than gloss words alone. The success achieved by using glosses for query expansion is consistent with the approach that will be shown in SemSeT experiments, where non-inheritance based expansions (referred to as relation classes) produced improved results.

A recent survey, Bhogal (Bhogal et al., 2007) said that using ontology is problematic because the success of *OQE* has some key dependencies:

- the quality of an ontology, knowledge model or thesaurus is paramount;
- a successful search process is more likely if the user is familiar with the ontology;
- the ease of the user's query process can improve search effectiveness, e.g. where users are able to navigate an ontology more easily because the search process automatically suggests expansion terms - so that the user can choose relevant terms.

The issue of managing interface complexity was also considered in SemSearch (Lei et al., 2006) and is consistent with the approach that has been adopted in the SemSeT process, where the user is required to manage context selection and related concepts are then presented for selection. Bhogal et al. discuss the issue of word ambiguity and refer to the problem of vocabulary mismatches between the query terms and the concepts in the ontology, although make the point that ontology offers a solution for word sense disambiguation. A differentiation was made between domain-independent and domain-specific "ontologies", with WordNet's domain-independent broad coverage considered likely to present problems of ambiguity; not surprisingly, it was suggested that domain-specific ontologies are preferable for narrower search tasks, given that terms and concepts are more likely to be accepted in a given domain. However, the possible absence of any required domain ontology was highlighted as a problem, on the basis of ontology development effort required, i.e. knowledge extraction from domain experts and achieving a consensus view. This point is interesting because two of the SemSeT experiments were based on rapidly developed, small ontology modules that did not involve considerable development time or domain expertise. Further, the issue of having gaps in the menu of ontology contexts available for selection is inevitable when developing a new search strategy and simply a matter of scaling up resources over time.

1.7 ONTOLOGIES FOR SEARCH CONTEXTS AND REUSE

It should be noted that this section, in the main, forms part of the research and not part of the literature review as such. It was considered appropriate to address some ontology development issues at this point because it amplifies some reuse issues discussed in the literature review conclusions in section 1.8. As the research examines query expansion using corpus independent knowledge models this section provides part of the contribution referred to in section 1.9, i.e. related to as "concept duplication (redundancy)". It is concerned with how modular, self-standing OWL ontologies (to be termed contexts) could be developed for integration and reuse purposes; and used for *OQE,* in a prototype semantics-enabled search tool.

1.7.1 Ontology for Purpose

This research proposes that self-standing ontology contexts can be developed to support the principle of ontology reuse, reduce processing overhead (e.g. caused by any concept redundancy resulting from clustering ontology modules), and to support query expansion. The focus will be on refining modular ontologies, in that modules containing duplicate concepts at the *outset* will be rationalised to make them contextually disjoint, i.e. more specialised. The discussion involves examples of transportation sub-domains related to road and rail transportation.

The owl:imports construct allows reuse of existing concept definitions, to give an importing ontology O_a a contextual relationship with the import O_b. Imports are useful because owl:imports statements are transitive, i.e. if O_a imports O_b and O_b imports O_x, then O_a imports O_x. However, what happens if we want to say that a Rail ontology concept RailOperator (shaded yellow in Fig. 22) is a kind of Cyc (Lenat, 1995, Cycorp, 2005) TransportationCompany?

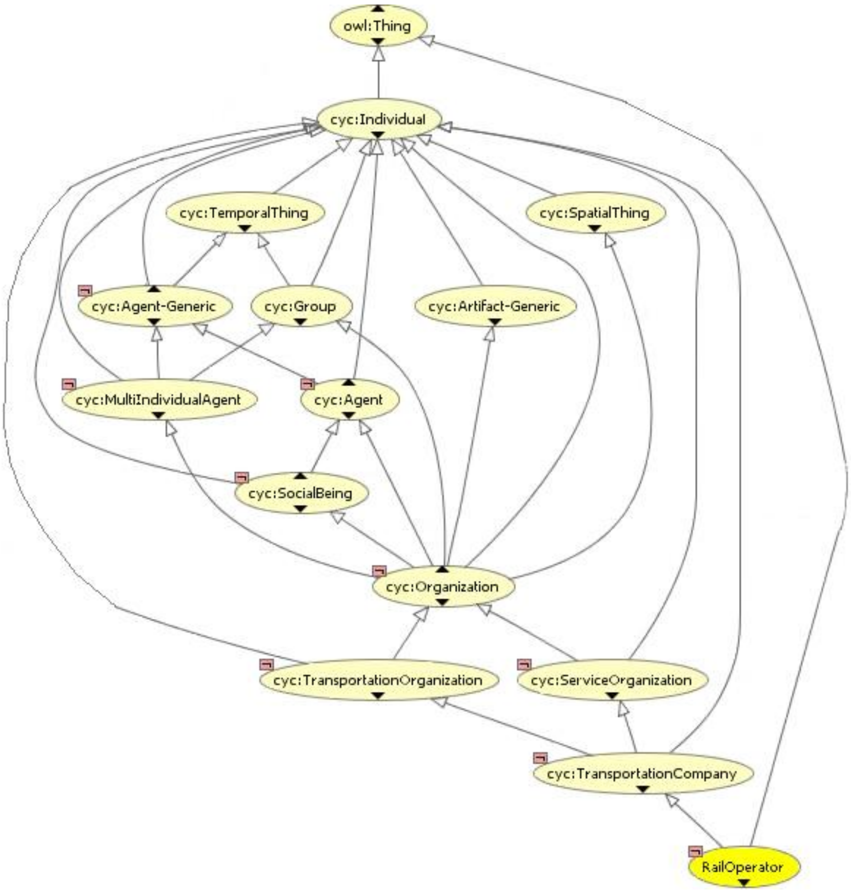


Fig. 22. Immigration classes mapped to SUMO classes. (For schematic representation only)

By importing say OpenCyc into Rail, the construct copies not only the TransportationCompany super classes but also the full hierarchy and axioms of OpenCyc, even though we may have no use for them. Therefore OWL has a weakness in that it either permits access to all foreign axioms or none, because all axioms in the joined ontology must be satisfied - referred to as “global semantics” (Bouquet et al., 2003).

However, using a seemingly relevant and smaller ontology can also create reuse issues, e.g. if we were to import the Ontology of Transportation Networks (OTN) (Lorenz et al., 2005) into our transport/tourism theme, we would find multiple sub-domains. Fig. 23 shows an abstraction of the ontology with a range of potential sub-domain redundancy: Land_Cover_and_Use; Road_and_Ferry; Meteorology; Railways; Service; Education; Entertainment; Public_Buildings; Tourism; Emergency; Food_and_Housing; Shopping.

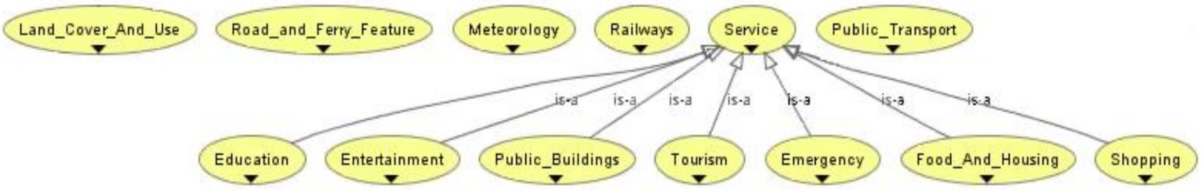


Fig. 23. An abstraction of sub-domains contained in OTN.

A contextually relevant import, but demonstrating design and development autonomy, would result in inherited complexity and processing overhead because using the OTN in a query expansion based on the *whole* ontology, would require a full traversal of the ontology to identify potential concept matches against query terms. This processing overhead would be better controlled, by having a user facility to selectively include relevant ontology contexts and their concepts prior to query execution.

The issue of selective concept reuse has been examined using E-Connections (Grau et al., 2006) and could be a solution as it allows specific concept linking to foreign ontologies via OWL language extensions; but could it be managed to generate a specifically targeted module from within an existing ontology? Nevertheless, the approach suggests that "small components" provide flexibility and usability. However, would it be better to design modularity at the conceptualisation and design stage?

1.7.2 Designed Modularity for Reuse and Minimal Redundancy

Ontology modularisation initiated at the conceptualisation stage can be considered a design *choice*, i.e. it provides inherent characteristics for reusing modules as library/menu items, e.g. in this research as search ontology contexts. Conversely, formalisms created as in E-Connections and C-OWL (Bouquet et al., 2003), really represent post-design, module extraction *remedies* that are independent of original ontology design and development decisions; however, their approaches could provide a solution for selective reuse of ontology concepts, for incorporation within a query topic relevant ontology module/context. E-Connections requires participating ontologies to be disjoint and provides no facility to create subsumption classes between the ontologies but this would not preclude semantically related foreign concepts from being incorporated into query ontology context, e.g. when constructing the TREC topic ontologies discussed later.

Ontology contexts are proposed as one solution for flexible reuse, processing and minimising redundancy (*low cost*), and are considered suitable for a search tool like SemSeT. A designed modularity approach should be viewed as in effect *de-integrating* a domain module at the conceptual stage, so that independent (self-standing) and almost disjoint contexts can be created and *re-integrated* to satisfy contextual search; this will be examined further by:

- considering visualisation and scoping of domains of interest (subsection 1.7.3) and how ontology sub-domain modules might typically be designed (subsection 1.7.4);
- understanding the issues resulting from module grouping/clustering (subsection 1.7.5);
- re-conceptualising and developing modules as disjoint contexts for queries to minimise redundancy (subsections 1.7.6 and 1.7.7).

A mix of concepts from the road and rail transportation sub-domains will be used to demonstrate redundancy and examine how modules can be specified more efficiently and effectively, to provide query flexibility in ontology context selection by:

- minimising future rework: i.e. avoiding having to revise a specification, as ontology stability contributes to reusability; developing a durable ontology by focusing on primary concepts, i.e. concepts that are contextually restricted to an ontology;
- minimising potential redundancy: avoiding redundant terms across ontology modules - to reduce potential mappings and minimise query complexity and processing overhead;
- using a consistent, best practice approach for meaningfully describing concepts: their relationships and constraints based on (Rector, 2003) – to facilitate search processes by providing primitive, stand-alone/atomic classes for use as query expansion candidates.

The aim would ultimately be to have a menu of ontology contexts to support a semantics-based search tool, where any clustering would deliver a *low-cost* group of otherwise disjoint (semantically unrelated) sub-domain modules (contexts for queries). This approach clearly differs from semantic similarity (semantically related) clustering to overcome heterogeneity (Ding and Foo, 2002).

1.7.3 Scoping Ontology Modules by Visualisation

A single general transportation ontology could support various search applications, e.g. passenger travel, freight services, tourism, transport communication systems, etc. Whilst all relevant concepts can be described in a single ontology, would selecting such an ontology offer effective reuse? For example, a tourism application would have little use for freight services and what if tourism only required ontology support for rail-air systems? Clearly, reusing the ontology for tourism could mean concept redundancies; so, is modularity more appropriate?

For this discussion, the ontology transport domain will comprise road, rail passenger and freight transportation, serving population centres; but how can a model of this mix be effectively and efficiently described? In terms of visualisation, consider the road/rail/population centre sub-domains represented in the UK's South East land transport system connecting towns/cities – see the schematics in Fig. 24.

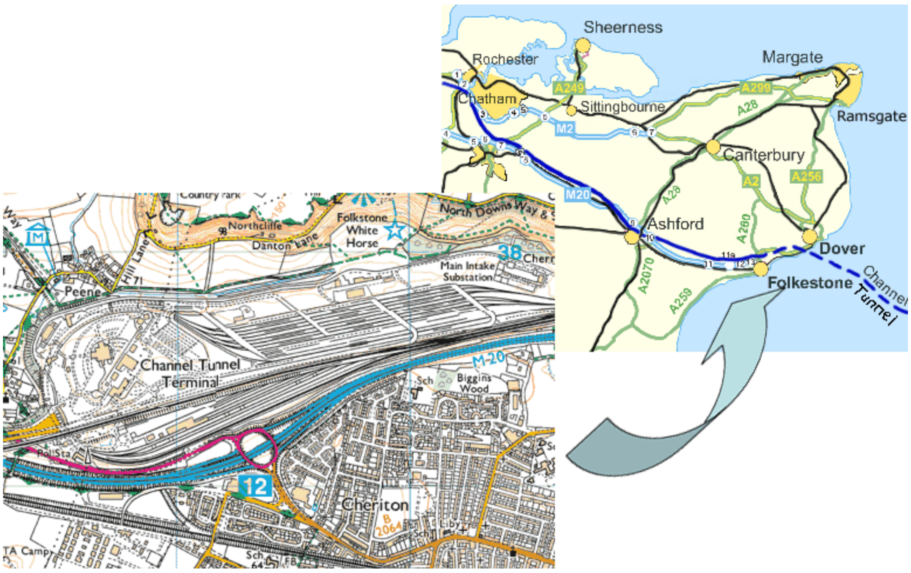


Fig. 24. South East transport and CTRL Terminal* - © OS Get-a-Map*

These schematics depict roads, motorways and railways plus the Channel Tunnel Rail Link (CTRL) - essentially a single mode of transport interfacing with road transport. Other road-rail interfaces might be level crossings and multimodal transport interchanges, e.g. the drive-on, drive-off service at Cheriton CTRL road-rail terminal. These schematics serve as visualisations for conceptualising and describing a cluster of ontology modules.

For the purpose of clarity, a transport sub-domain will now be referred to as a module.

1.7.4 Module Conceptualisation and Design

A road and rail transportation ontology model can be viewed as a multimodal system encapsulated by logical modules (Road, Rail, PopGroup), e.g. the Rail module could be described using some simple Protégé-developed object property statements, which are reflected in the Rail model in Fig. 25 (a).

RailRoute startsFrom (RailwayStation $\sqcup$ City)
RailwayStation locatedIn City
RailRoute hasRailComponent RailwayLine
RailwayLine meetsObstacle LevelCrossing
LevelCrossing intersectionBetween (RailwayLine $\sqcap$ Highway)
RailwayStation accessedVia Highway
Highway startsFrom (RailwayStation $\sqcup$ City)

Similarly, as shown in Fig. 25 (b), the Road module might say, "highway provides access to a city and a CTRL terminal"; "terminal offers a drive-on/drive-off facility and is accessed by rail"; "highways encounter railway lines and level crossings". However, in describing the Road module, it is evident that certain concepts (City, Highway, LevelCrossing, RailwayLine) are duplicated across Rail and Road.

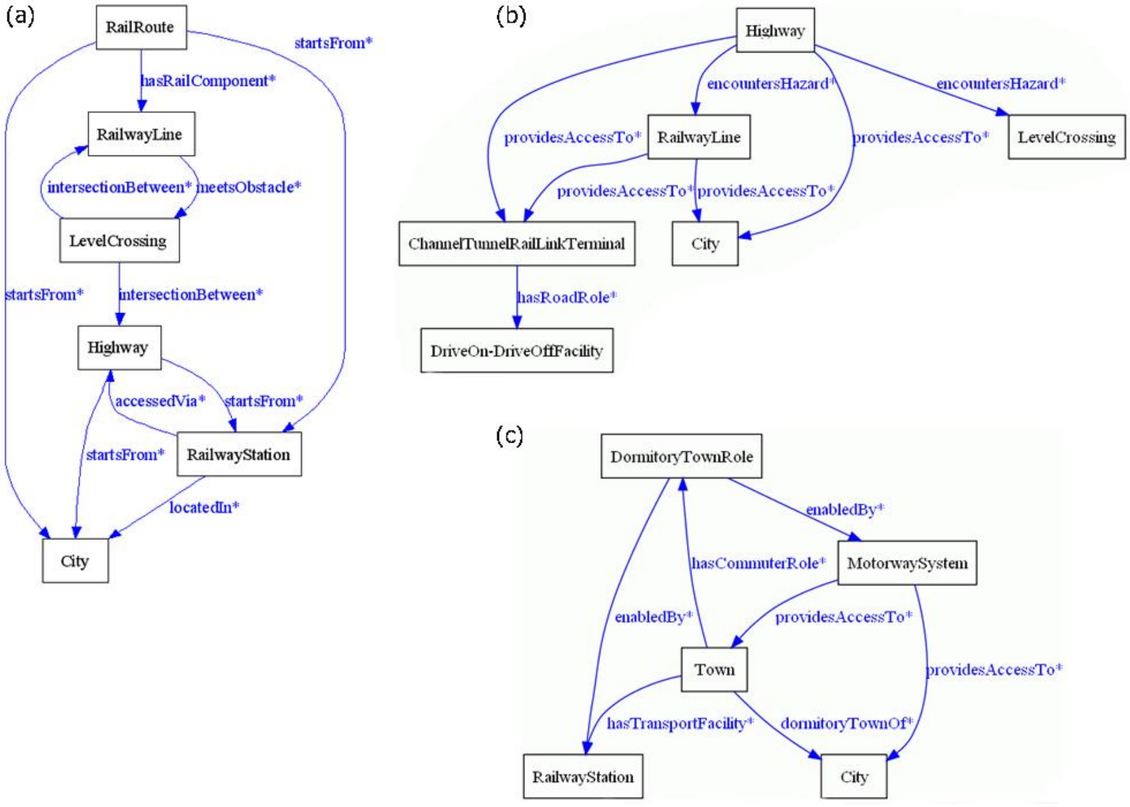


Fig. 25. Models of (a) Rail, (b) Road and (c) PopGroup ontology modules.

Finally, Fig. 25 (c) shows a PopGroup fragment describing possible classes and relationships between City, Town and a DormitoryTownRole "enabled by" a MotorwaySystem and RailwayStation. Again, we find concepts like City and RailwayStation have been duplicated.

1.7.5 Clustering Modules for a Multi-context Ontology

The three modules can be clustered by importing them into a Land Transport application *general* ontology. Let us assume the general ontology, shown in Fig. 26, has its own concepts, i.e. it contains general and multimodal transport concepts, e.g. TransportInterchange, TravelCentre, TransportOperator, and some transport relations. These could serve as semantic anchors for imported module concepts, e.g. we might say that a TravelCentre is located in a RailwayStation or that a TransportOperator operates from a ChannelTunnelRailLinkTerminal.

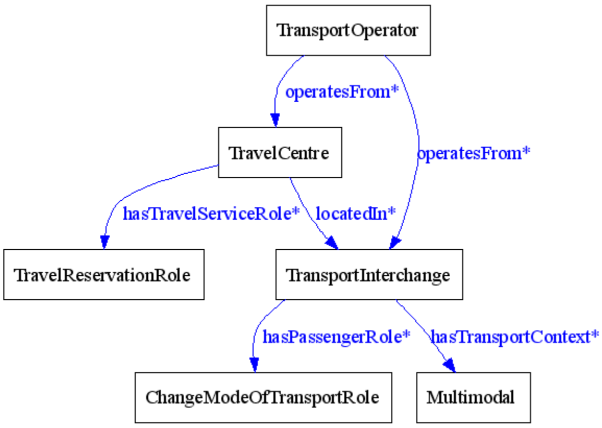


Fig. 26. Model of Land Transport concepts and relations.

But, what are the implications of importing Road, Rail, and PopGroup modules into the general Land Transport ontology and specifying new relationships? The result of this multi-context clustering is shown in Fig. 27, with Land Transport general and multimodal concepts differentiated by shading (in yellow) and now with the following object properties applied:

TransportOperator operatesFrom (ChannelTunnelRailLinkTerminal ⊔ RailwayStation)

TravelCentre locatedIn (ChannelTunnelRailLinkTerminal ⊔ RailwayStation ⊔ City ⊔ Town)

However, to achieve this, imported class namespaces are required in statements, e.g. rail:RailwayStation, road:City, and this highlights various issues in the general ontology.

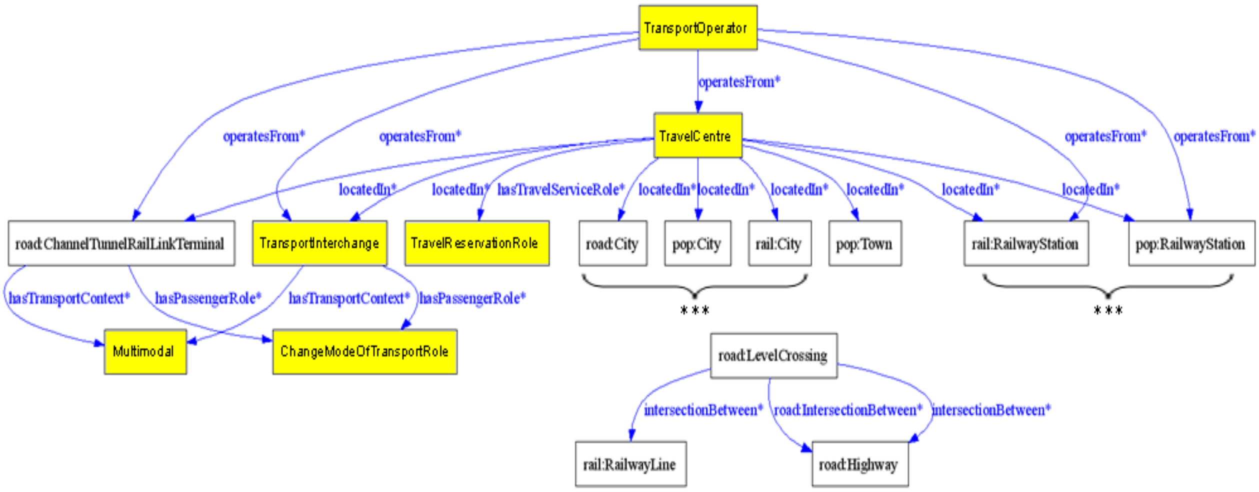


Fig. 27. Redundancy resulting from duplicated classes in Land Transport.

For ease of presentation, not all relationships specified earlier in Fig. 25 are shown in Fig. 27 but this simplified model demonstrates concept duplication and redundancy (duplications are denoted by ***) between rail:RailwayStation and pop:RailwayStation, and with rail:City, road:City and pop:City. If all concepts and relations had been shown there would have been duplications in classes Highway, LevelCrossing and RailwayLine and relation providesAccessTo. Any new relations between Land Transport classes and imported classes must be specified in Land Transport, as imported ontologies retain autonomy, i.e. imported *local* object properties, domains and ranges will endure and create potential redundancy, e.g. consider an object property linking Road and Rail classes: road:LevelCrossing intersectionBetween (rail:RailwayLine ⊓ road:Highway); whilst an axiom can be created in Land Transport, any *local* specification of domain and range requires a new object property intersectionBetween, thereby duplicating the road:intersectionBetween relation in Fig. 27. Clearly, integrating modules that are not wholly disjoint creates an overhead and, with multiple semantically related classes, requires mappings, e.g. equivalence - adding further complexity. Therefore, a more streamlined or partitioned design is suggested, to progress from structural modularity to semantic modularity (ontology contexts), i.e. reflecting semantic modality.

1.7.6 Re-Conceptualisation and Specification of Disjoint Contexts

How might the earlier transport-related modules be re-conceptualised and designed as contexts? The earlier Fig. 24 schematic of the multimodal Channel Tunnel Terminal hides the physical and semantic modality between Road, Rail and PopGroup, which is revealed when re-visualised as separate geographical layers - see Fig. 28; a metaphor for this would be map layers that are subsequently combined to represent topographic features. Fig. 28 serves as a vehicle to conceptualise and specify disjoint modules and minimise redundancy.

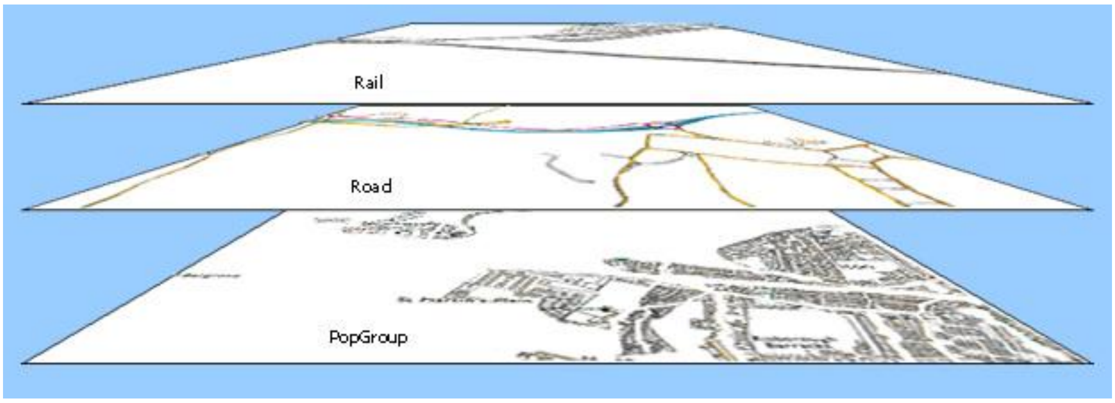


Fig. 28. Separation of combined context schematic of Rail, Road and PopGroup.

This can be applied in other domains, e.g. in utilities where semantic layers can differentiate gas, water and electricity systems for say planning applications. In effect, semantic layering suggests a conceptual process of module de-integration to make several context distinctions.

In an approach to minimise reuse redundancy, Road world and PopGroup world concepts should not be described in Rail world, and vice-versa, e.g. we should say a RailRoute can only start from a RailwayStation and not a City; similarly, a RailwayStation can not be accessed by a

Highway, which may start from a RoadJunction but not a City. By using only *primary* concepts when specifying relationships in a context, we say that, in Rail, a start-point City (a primary concept in PopGroup) is secondary to RailwayStation. To explain this further, assume the Rail module is symbolised by the model in Fig. 29.

As a preliminary explanation of the model, primary context relations are shown as solid edges, e.g. R_{P1}, which links primary context (Rail) CT_P domain class C_{P1} and primary context range class C_{P2}. A secondary context relation (serving to link between *only* secondary context concepts) are depicted with dotted edges e.g. R_{S3}, which links secondary context (Road) CT_{S1} domain class C_{S1} and secondary context (PopGroup) CT_{S2} range class C_{S2}.

Relations serving *only* to link between primary and secondary context concepts are also depicted with dotted edges but are distinguished with ~, e.g. $R_{P\sim1}$ and $R_{S\sim1}$. However, relation R_{P1} linking Rail concept C_{P1} and PopGroup concept C_{S2} is different, as it already exists as the relation linking Rail concepts C_{P1} and C_{P2}.

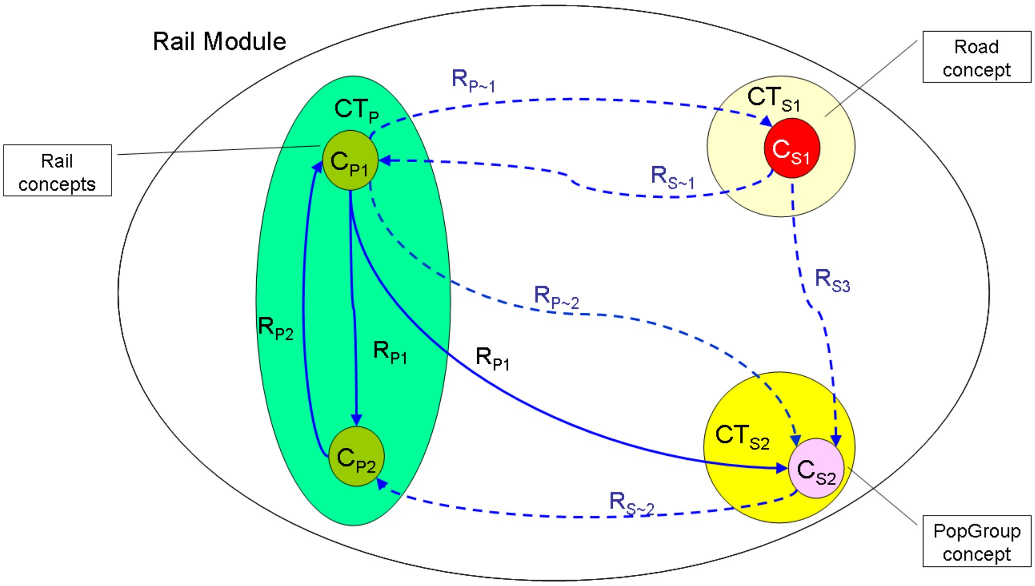


Fig. 29. A model of multi-context relationships contained in Rail module

Using the Fig. 29 model, the general approach can be presented more formally.

> Let ontology module O that contains *classes* C, *relations* R and has a *context* CT be a set $O = \langle\langle C_{(1,,,n)}\rangle, \langle R_{(1,,,n)}\rangle, CT\rangle$. Further, let any concepts represented in that ontology be shown as either *primary classes* $C_{P(1,,,n)}$ or *secondary classes* $C_{S(1,,,n)}$, their primary and secondary *relations* as $R_{P(1,,,n)}$ and $R_{S(1,,,n)}$ respectively, and the primary context as CT_P and the secondary

contexts $CT_{S(1,,,n)}$. A multi module represented ontology set can then be shown as:

$$O = \langle\langle(C_{P1,,,}C_{Pn}),(C_{S1,,,}C_{Sn})\rangle,\langle(R_{P1,,,}R_{Pn}),(R_{S1,,,}R_{Sn})\rangle,\langle CT_{P} ,(CT_{S1,,,}CT_{Sn})\rangle\rangle$$

Using this approach, the following de-integration rules should be applied to Rail:

i. if a relation has only primary classes in its (object property) domain *and* range then the relation is termed a primary relation, e.g. $R_{P1}(C_{P1,}\ C_{P2})$ and $R_{P2}(C_{P2,}\ C_{P1})$;

ii. if a primary relation *also* specifies a secondary class as range, the relation remains primary, e.g. as in $R_{P1}(C_{P1,}\ C_{S2})$;

iii. if a relation will *only* have secondary classes as domain *and* range then the relation is termed secondary, e.g. R_{S3};

iv. if a relation is not primary or secondary, the relation's domain class will determine relation context (if retained), e.g. primary class as domain $R_{P\sim1}$, $R_{P\sim2}$ and secondary class as domain $R_{S\sim1}$, $R_{S\sim2}$;

v. for each secondary context module remove their "secondary" classes and relations – they will be primary in their own contexts.

These rules address most situations, except that Fig. 25 (a) and (b) show that LevelCrossing is relevant in both Road (as rail crossing) and Rail (as road crossing). So how could this be addressed, given a *single context* and unimodality is sought for each module, as LevelCrossing is clearly multi-transport contextual, i.e. multimodal? Therefore, module *multi-context* concepts are *elevated* to the generalised *multimodal* application level, i.e. LevelCrossing is removed from Rail and Road, as a secondary class and specified as primary in Land Transport; the same applies to ChannelTunnelRailTerminal. Equally, in Land Transport, unimodal concepts would be specialised to relevant contexts. Rules for classifying linking relations similarly apply.

The above can be viewed as a qualitative, *pre*-specification partitioning approach requiring intuitive understanding, as opposed to a *post*-specification structure partitioning approach, as in (Stuckenschmidt and Klein, 2004), that is quantitative and relies on measurement.

Interestingly, the Ordnance Survey defined *core* and secondary concepts in their ontology development methodology (Hart et al., 2007), where a topographic domain includes concepts Road, River, Hill and Building as core (i.e. within the scope of the domain) but treats Water (e.g. River transports Water) as secondary, as it is not essential for topography. This is very similar to the de-integration approach presented here, except that here any three of Road, River, Hill and Building would be identified as secondary concepts and placed in separate contexts to support potential reuse.

1.7.7 Results of Designed Modularity

The effect of designed modularity on the earlier Land Transport model, in Fig. 27, is shown in the revised model Fig. 30, i.e. each class is now specified in its primary context only, with secondary duplicated concepts removed.

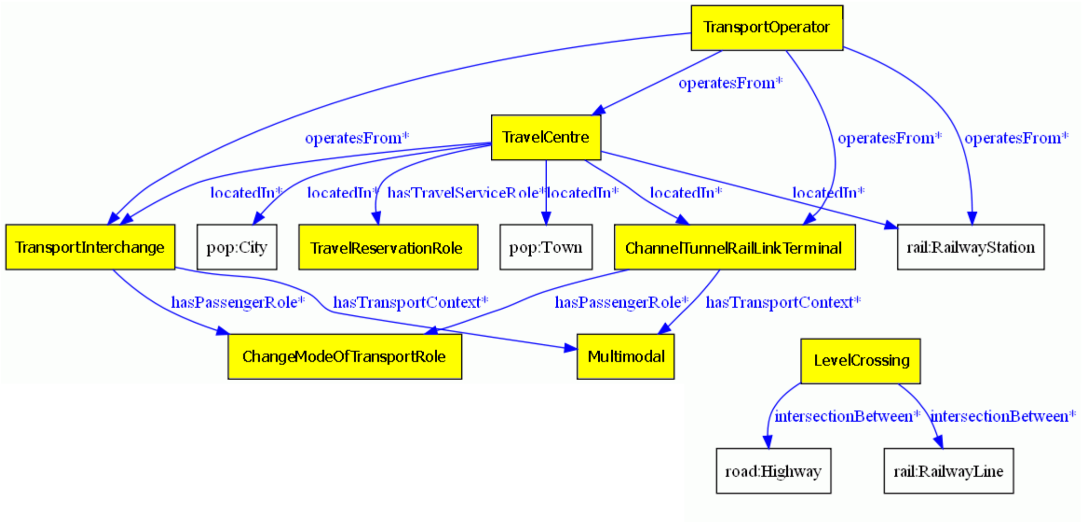


Fig. 30. A revised Land Transport ontology model with duplication removed.

A comparison with Fig. 27 shows class duplication has reduced markedly. When duplications not shown in Fig. 27 (for Highway, LevelCrossing and RailwayLine) are included, classes are reduced by a third and the number of relations is also reduced. Each of Fig. 25's modules characterised the set: $O = \langle\langle(C_{P1,,,}C_{Pn}),(C_{S1,,,}C_{Sn})\rangle,\langle(R_{P1,,,}R_{Pn}),(R_{S1,,,}R_{Sn})\rangle,\langle CT_{P} ,(CT_{S1,,,}CT_{Sn})\rangle\rangle$ but each module has now been returned to $O = \langle\langle C_{(1,,,n)}\rangle,\langle R_{(1,,,n)}\rangle,CT\rangle$, ready for importing. However, an accepted issue is that any module's original primary to secondary class relationships, e.g. axioms or domain/range, would likely have to be re-created between the imported primary-to-primaries within Land Transport, although only as required by the general application ontology.

How can this approach be used in semantic search? A number of contexts were created for the TREC *OQE* experiments, and were used both as individual contexts and integrated contexts, e.g. one experiment used a wider Tourism ontology embracing over 650 concepts through various imports, and two experiments used smaller bespoke contexts - the Immigration and Hydro-electric ontologies; these are discussed in chapters 3 and 4.

1.8 LITERATURE REVIEW CONCLUSIONS

The literature review sections, together with section 1.7, have provided a broad understanding of how a corporate and consumer society has to contend with information integration and search issues over a period of three decades. In the last 10-15 years there has been a semantics-driven progression towards information integration in the "global information space" of the Web, e.g.

information brokering systems using ontology; therefore, it seems logical that extending query semantics, by using ontology, will be predominant in search-driven “integrations” of information sources. And yet, the Semantic Web community only recently appears to have become really focused on Semantic search and there are no significant examples in the public domain. Further, search engines provide little evidence of exploiting developments in the Semantic Web, i.e. exploiting the increasing availability of machine readable documents provided with contextual relevance through ontology definition - probably because of the size of task in creating sufficient RDF resources (linked data Web) to mirror traditional (unstructured) Web documents and, in any event, why would they not wish to continue searching the traditional HTML document Web? Google have recently begun to use synonyms [1] but there is no evidence of ontology usage [2].

A semantic search tool should not necessarily restrict a user to making queries on RDF/triple-based data repositories; the tool should be able to exploit the mass of existing (unstructured) Web documents that will inevitably have no metadata annotations or semantics links. Effective IR depends on the capability to return contextually relevant documents and ontologies are designed to formally specify the shared contextualisation of domains; these two capabilities provide the basis for a mutually beneficial approach. Therefore, can Semantic Web technology be applied to the traditional document Web, as opposed to the Semantic (Linked Data) Web?

1.8.1 Ontology-based Query Expansion

How would ontology be used to enhance Web document search and what impact could it have on the accepted measures of P&R? What tool would be available to do this? It is unlikely that any experimentation would be able to exploit existing search engine capability as uncontrollable variables might apply, i.e. specific search engine algorithms, e.g, Google’s PageRank (Google, 2008); it would not be possible to integrate an ontology into their *OQE* process; search engines are selective as to which pages they return. Further, it would be logistically impossible to verify the true relevance of returned documents and would require independent verification. The challenge would be to use a Semantic search tool to conduct comparison query experiments, although, how might it work? Some query expansion approaches have required term disambiguation before selecting query expansion sets (Voorhees, 1994).

One issue must be the degree of concept propagation in query expansion, e.g. how far up and down an ontology class hierarchy should traversal progress? During the search process, how could the hierarchical levels of class relevance (to a query topic) be determined for any given ontology having greater or lesser generalisation, specialisation or complexity?

[1] http://www.mattcutts.com/blog/google-synonyms/.

[2] http://googleblog.blogspot.com/2010/01/helping-computers-understand-language.html.

One simple and practical solution could be to limit the hierarchy context by creating multiple ontology contexts. Parallels exist, as probabilistic models and search engines make use of term clustering, e.g. the Google "Suggest" functionality. Therefore, it should be feasible to identify the most popular search topics and build a menu of query topic relevant ontology contexts. The challenge then becomes one of scaling up the menu of context choices and finding a mechanism to manage them via a combination of user interaction and process automation.

How would a user conduct *OQE* and how would an appropriate ontology context be identified? How might it be led, e.g. should the user be expected to construct complex structured queries or should the system functionality support the user with application algorithms? User query formulation often requires understanding of structured query languages, however, for majority of users a public semantic search tool will need to offer the simplest and most understandable process to assist the user with semantic tool functionality. From a practical user perspective it is considered that small ontology contexts could be more navigable during the query input and could therefore ease the *OQE* search process, i.e. user handling of small ontology contexts, for both context and query term selection, could be assisted by incorporating assistive algorithms in the tool to demonstrate how query relevant contexts might be identified to guide users in selecting query relevant ontology context terms - see subsections 3.1.3 to 3.1.5, regarding query input interface and state transition network (STN) diagram.

1.8.2 Ontology Modularity and Contexts

How should ontology be used - generalised and large versus discrete and contextually specific? How best might they be constructed - less emphasis on hierarchy and more on asserted conditions or axioms? Section 1.7 argued the case for developing modular, self-standing ontology contexts for *OQE*. IR can only be successful when the subject context is reflected in the query term context; equally for *OQE* to be search effective the query context should be supported by a matching ontology context. Therefore, based on section 1.7, many ontology contexts would be required to support different types of search topic, which will raise an issue of how the user would be able to manage the ontology context selection process.

1.8.3 Algorithms for Determining Document Relevance and P&R

The vector space model (VSM) and the probabilistic models (PM) were considered for this research. VSM is designed to deliver either retrieval or non-retrieval based on known terms, whereas PM is based on ranking documents by estimating the probability of relevance. It is generally accepted (Cleverdon, 1991) that effective IR systems should optimise the number of all relevant documents in a retrieved set (recall) and minimise the number of non-relevant documents (precision). Therefore it is appropriate to base search tool effectiveness on precision and recall.

1.8.4 Impact of Semantic Search

In terms of outcomes, would *OQE* result in lower precision and greater recall, or vice-versa? Would semantic search inevitably make traditional keyword search moribund? Should the user have the choice of either or both, e.g. queries might start with keyword method and ontology-based search might be used as an option if keyword fails? Reference has been made about improving recall, e.g. by returning documents containing non-keyword terms, which may be none-the-less query relevant? How are these to be identified and also would they materially add value to any returned ranked hit list, bearing in mind that Web search users are unlikely to make use of more than the first few pages of potentially relevant hits returned by search engines?

1.8.5 Semantic Correlation between Ontology Concepts

The semantic correlation between ontology concepts will have a bearing on the degree of relevance that should be attributed to any such concept relative to a base (original) query term. The traversal of a subsumption hierarchy, including semantic relations specified to describe and constrain classes, will clearly determine how and to what extent an expanded query term set will be developed in an ontology-based query expansion. An equally important consideration is the process by which different relationships would be taken in account in any document relevance measures (algorithms), when relevant, or semantically related, expansion terms are generated from matched query terms. This can be achieved by reflecting the semantic distance from a base query term by weightings and this has been considered and demonstrated in various ways (Fang et al., 2005, Gligorov et al., 2007, Tiun et al., 2001, Rocha et al., 2004, Bhogal et al., 2007); where different weightings are used to differentiate the type of semantic relationships.

The size and domain coverage of an ontology will also have a bearing on relevance; a more course-grained (top-level or domain) ontology tends toward generality, imprecision and abstraction; which, although more *shareable* to wider domains and applications, may be less useful because of lower expressivity (Bhogal et al., 2007). However, as was discussed in section 1.7, concept duplication can easily occur when reusing ontologies for *OQE*, which could present problems in terms of processing overhead and risk of duplication, e.g. with reuse of overlapping ontologies (subsection 1.7.1) or with extensive use of class asserted condition relations (subsections 3.2.3 (Pseudo Code for Relation Class Algorithm) and 3.2.5). For this reason, the degree of relevance and size of an ontology needs to be considered when conducting *OQE*. How this will be addressed is demonstrated in chapter 3, Experimentation.

1.9 PROBLEM STATEMENT

The purpose of this PhD is to use *OQE* to improve search effectiveness by increasing search precision, i.e. retrieving relevant documents in the topmost ranked positions in a returned document list. Query experiments have required a novel search tool that can combine Semantic Web technologies in an otherwise traditional IR process using a Web document collection.

The research will address two contributions to knowledge, the first concerns how modular, self-standing OWL ontologies (called contexts) could be used in ontology-based query expansion (*OQE*), in a prototype semantics-based search tool developed for the experiments. The second examines how the search tool could manipulate such Semantic Web-based *OQE* to improve information retrieval (IR) search effectiveness, compared to traditional keyword-only search, on ordinary HTML documents; i.e. as opposed to the predominant current research of using semantic reasoning-based RDF query languages on Semantic Web triple repositories, to refine the query process automatically. Therefore, the distinction is that Semantic Web technology would be applied to the traditional (unstructured/semi-structured) Web, as opposed to the Semantic (linked data) Web. Integral to the use of ontology will be how to facilitate reuse with minimal concept duplication (redundancy) and processing overhead, when ontology contexts are combined; section 1.7 addressed this element of the problem statement. Impacting on these issues will be the practical problem of how to simplify selection of ontology contexts and their candidate *OQE* concepts.

1.9.1 Research Challenge

The primary objective will be to improve relevant document rankings, i.e. increase IR precision and improve search effectiveness. The return of additional relevant Web documents (for recall), e.g. those containing none of the base query terms, would be a secondary benefit.

To support Semantic Web-based *OQE* and improve search effectiveness, the research experimentation requires a large document corpus, query relevant ontologies, a query interface, a keyword and ontology traversal text matching mechanism in a prototype search tool, supported by document ranking algorithm to facilitate keyword versus *OQE* search relevance comparisons.

1.9.2 Hypotheses for Issues Identified

The query experiments will be used to test the following research hypotheses:

i. hierarchical *OQE* can have a positive impact on precision and recall, although class hierarchy expansions alone may not produce optimal results. Query term-matched classes may have more beneficial wider semantic *relations* with other classes, beyond simply super and sub class hierarchies, and exploiting the expressivity of the OWL ontology language, by using asserted conditions, will provide useful *OQE* options and improve document relevance scoring and ranking. This will be tested by comparing search effectiveness of keyword-only query against various *OQE* modes - see T401 and T416 experiments in chapters 3 and 4;

ii. higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes identified in the *OQE* process; this would preserve the importance of the original (base) keyword input by the user and reflect the semantic distance between the base keyword terms and their expanded terms. This will be tested by comparing queries using weighted and non-weighted *OQE*s – see subsections 3.2.5 and 4.4.1 to 4.4.3;

iii. topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion despite their potentially restrictive coverage, i.e. they can still capture the essence of a (TREC) query topic and improve precision and recall, as opposed to contextually wider or more comprehensive ontologies; i.e. the emphasis should be on restricting ontology size, to avoid superfluous query expansion. This proposition will be tested by controlling the extent of *OQE*; by creating small, self-contained (restricted and flatter hierarchies) query topic relevant ontology contexts and comparing the P&R results, of various *OQE* modes, against contextually wider or more hierarchical, larger ontologies - see T401 and T416 experiments versus T438 experiment (chapters 3 and 4) and T401 versus *extended* T401 (subsection 4.4.4).

The ability to evaluate the hypotheses will be dependent upon the following questions.

i. Has an impartial and unbiased search comparison process been employed?

ii. Does the search tool support ontology traversal and relevance ranking mechanisms effectively and reliably?

iii. How useful were ontology query contexts, e.g. concept usage?

iv. Did the results show meaningful improvements in either precision or recall?

2 RESEARCH EXPERIMENTATION APPROACH

This chapter provides background to the research experimentation approach in establishing how a search tool could exploit Semantic Web-based *OQE* to improve IR search effectiveness; compared to traditional keyword-only search on ordinary HTML documents. The experiments will not employ semantic reasoning-based RDF query languages on Semantic Web triple repositories, to refine the query process automatically; therefore, the distinction is that Semantic Web technology would be applied to the traditional (unstructured) Web, as opposed to the Semantic (linked data) Web.

Envisaged benefits could be in improving relevant document rankings (for precision) and in returning additional relevant Web documents (for recall). However, as only a few pages of search engine results tend to be useful to Web users, query precision results are likely to be most indicative of meaningful search effectiveness in the early, low recall intervals. Therefore, the research experiments will measure the success of *OQEs* against keyword-only search by comparing precision outcomes, primarily in the 10% to 30% recall range, i.e. by comparing *OQE* P&R curve outcomes against the "base" keyword P&R curve profile.

2.1 METHOD FOR SEARCH EFFECTIVENESS MEASURE

To test the hypotheses proposed in subsection 1.9.2, a scientific approach will be used to evaluate IR, by comparison of *OQE* query outcomes against a control set of keyword query results. The experiments will be conducted using query subject relevant ontologies, ontology traversal, term matching and relevance scoring mechanisms, and evaluated using a P&R analysis, based on identifying the position of relevant ranked documents returned, to compare and determine their relative search effectiveness in document retrieval (subsections 1.6.2 and 1.6.3). The project will require a semantics-enabled search tool to conduct queries, in both keyword and *OQE* (semantic) search modes, and generate the relevance scores for subsequent P&R analysis.

To provide independence and experiment control, traditional search engines will not be used and a novel prototype semantic search tool (SemSeT) will be developed to facilitate the experiments. The tool will use Jena Ontology API methods, to traverse OWL ontologies and extract classes and instances, for *OQE* but the tool will not use the Ontology API inferencing capability to distinguish between asserted and inferred types.

As discussed earlier (subsection 1.6.1), ranked retrieval is generally accepted as a preferred method as it attempts to calculate the merit of a document in satisfying a query and this will be adopted. The tool will need to incorporate a means of quantifying and storing document and term relevance, using established retrieval measures and the vector space model (VSM) and the probabilistic models (PM) were considered for this research; Ontology is based on formally specifying the vocabulary of a domain and therefore, given that query term expansion could justifiably involve the user in selecting a contextually relevant ontology, higher numbers of correlations between the user's term and the selected ontology hierarchy should indicate a strengthening relationship between the document text and the ontology context. Therefore, as the experiment seeks to compare the use of keywords against *OQE,* and not to justify VSM over PM, the choice between VSM and PM is not critical; therefore VSM *tf-idf* was selected for this particular research.

The experiment will require an independently verified document set, e.g. a document corpus that, for certain queries, has identified (known) relevance outcomes. In 2000 the TREC-8 Web Track (Hawking et al., 2000) featured, in its *Small* Web retrieval task, a ¼ million document subset distributed on DVD as the WT2g collection. WT2g comprised of a set of 50 TREC-8 Ad Hoc query topics that were each supported by a query requirement, in the form of a topic statement, and a set of query relevance judgements, listing a topic pool of documents distributed randomly across the full document collection. The density of relevant documents in the pool was approximately 0.92%. This collection will be used in the experiment. As indicated in section 1.9, the research project will involve conceptualisation of ontology models, for selected TREC query topics, and formal specification for various *OQE* mode experiments.

2.2 ENABLERS FOR EXPERIMENTATION

The methodology for delivering the SemSeT results involved a number of activities, including devising a search process and developing a search interface; identifying programming techniques for ontology traversal to extract ontology concepts and individuals; incorporating term relevance scoring and calculating *tf-idf* values for document ranking.

2.2.1 High Level Search Comparison Process

The flowchart in Fig. 31 essentially provides a high-level view of the key steps considered necessary to support SemSeT's keyword-only and *OQE* search, document relevance scoring and ranking process.

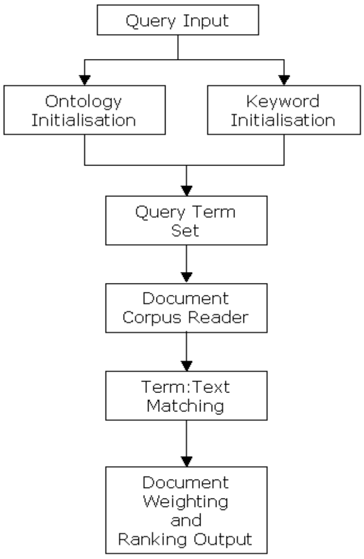


Fig. 31. High-level search process.

As the objective is to compare the search effectiveness of using *OQE* against keyword-only search, a set of query terms will be used for both keyword and *OQE* searches, i.e. each query comparison will be executed first in keyword mode, and then the same query term set will be reused for expansions in the various *OQE* modes.

2.2.2 Design and Development of Search Tool SemSeT

The primary purpose of SemSeT is to provide a prototype search tool, as an *OQE* engine to:

i. support various query expansion options, e.g. all ontology classes for a context-wide, general expansion (*All OQE*), sub classes only (*S OQE*), sub and super classes (*S+S OQE*), or sub and super classes plus relation classes (*S+S+R OQE*);

ii. generate statistics for comparing search effectiveness outcomes when using simple keyword search versus *OQE*.

The tool will also need to provide flexibility in setting the query term weighting conditions during the experimentation. Finally, SemSeT should demonstrate a practical way to assist a user in handling the semantic choices during *OQE* setup, e.g. an adaptive text algorithm will provide 'user field entry support' - for context and term identification and selection, by exploiting the tool's Jena Ontology API methods.

The Jena Ontology API's various iteration methods were considered sufficient for developing algorithms to traverse, query and extract OWL ontology classes and instances; therefore the inference API's capability to distinguish between asserted and inferred types was not used. Prior to development it was decided the tool should also have support for a *user* to be able to control query options using 'must' include and Boolean operators, like NOT and OR, although query term exclusions were not actually applied in the main experiments. It was also decided at the outset that document indexing would not be addressed, if there were insufficient time, given that search effectiveness was more important than search efficiency at this stage.

2.2.3 SemSeT Development Testing and Validation

To ensure reliability and validation of the tool and outputs, external black-box and internal white-box testing was conducted at each stage of development to verify the integrity of algorithms developed for: identifying relevant terms, documents, frequencies, *tf-idf* scores, and P&R statistics for search effectiveness analysis. The tool was first tested using a small test document corpus of some 100+ limited content online Web documents, which were created to provide a control set having pre-determined outcomes. When initial tests were satisfactorily validated, a sample of stored TREC data was used as a trial for the proposed formal experiments. As the TREC Web documents were concatenated in large text files, the tool was subsequently modified to handle the way TREC data was stored in folders; there were 28 folders, WT01-WT28, with most containing 40 sub-folders numbered B01-B40.

2.2.4 Procedures to Extract Ontology Concepts and Individuals

Based on potential user selections outlined in subsection 2.2.1, and the way OWL permits ontology class hierarchy specification, the *OQE* process needed to handle a range of ontology traversal issues: i.e. sub classes, sub and super classes, whole ontology, equivalent, intersection and union classes. A further requirement is to support relation class expansion where asserted conditions might exist. Algorithms are provided in subsections 3.2.3 and 3.2.4.

2.2.5 OWL Context Specification to Support *OQE*

The acquisition of TREC data provides known relevance outcomes across a range of query topics. After making searches for suitable ontologies on the Web (including using Swoogle), it became evident that bespoke ontologies would have to be conceptualised and developed for certain query experiments (i.e. T401 Immigration and T416 Hydro-electric) - consistent with the selected TREC query topics; the topics were used as the basis for the query ontology contexts. To ensure relevancy of ontology to query, it was decided to develop ontology modules as self-standing contexts to permit flexible clustering for contextual search, and reduce concept redundancy. Prior to the experimentation stage, some trial ontologies (Air, Sea and Tourism) were developed for testing the search tool during the development stage, in addition to Road, Rail and PopGroup ontologies developed for section 1.7. Protégé was selected to develop all ontologies in OWL DL format.

2.2.6 Term Relevance Weighting and Query Term Matching

As indicated in subsection 2.2.4, both the ontology class hierarchy and the axioms specified to describe classes will determine how an ontology-based query will gather related terms. Given the mix of relationships, to reflect both the semantic relations between the inheritance class hierarchy and other specified class relationships, different relevance weightings should be considered to reflect relevance against to any base query term.

2.2.7 Calculation of *tf-idf* Value for Ranked Document List

The handling of the *tf-idf* algorithm needed to be considered from two perspectives:

i. how the initial allocation of *tf-idf* component values would be stored during the document text and query term matching process, i.e. frequencies (for both relevant terms and relevant documents) and relevant term weightings, i.e. to produce key global *tf-idf* parameters.

ii. the subsequent manipulation of all *tf-idf* values to generate weighted document values - once key global *tf-idf* parameters had been identified.

The matching of each of the base query terms required a mechanism to store the *tf-idf* algorithm components, e.g. term frequency, document frequency for each relevant term in each relevant document until the complete document corpus has been interrogated. It was decided that manipulation could best be handled by storing the frequencies in separate arrays, created for the *OQE* terms list and the document list, so that the data could then be used in the *tf-idf* algorithm to derive ranked weighted document statistics – see example data in Appendix H.

3 EXPERIMENTATION

This chapter will examine three stages of the experimentation:

- the outline steps considered for the proposed experiments;
- how the experiments were designed;
- how the experiments were implemented.

3.1 SEARCH EFFECTIVENESS EXPERIMENT STEPS

This section sets the scene for the research approach by outlining the assumptions and steps considered essential to deliver the proposed experiments. It was decided that the main *OQE* experiments would be based on 3 TREC query topics, involving either 10 or 20 query term combinations (sets) per topic. Each query term set would be used to compare keyword-only mode to various *OQE* modes.

As the objective was to examine the impact of *OQE* compared to keyword search, on precision and recall (see hypotheses, section 1.9), it was decided to create a matrix of (TREC) meaningful queries by variously combining topic relevant query terms. This would result in a range of queries being executed over the document corpus, based on several query comparison options. The selection of query topics and creation of query matrices would be based on TREC query topic statements. The base query term combinations are set out in query matrices in section 3.3.

A comparison of *tf-idf* results, based on *precision* and *recall* (P&R) (van Rijsbergen, 1979), measures would then be made between the chosen query modes and plotted in a P&R graph (see subsection 3.1.7, Fig. 42 example). As mentioned in section 2, the assessment of query results will be focused primarily on precision outcomes in the 10% to 30% recall range.

3.1.1 Assumed User's Query Approach

It was decided that the approach for controlled query comparisons would be to start with a set of 4 base keywords or short phrases. This approach is simple and effective in basic Web search and keywords/short phrases can be more easily matched to ontology concepts and individuals.

3.1.2 Semantic Search Process

SemSeT queries should be executed by firstly entering up to 4 keywords/phrases in order to return pages containing either the keywords alone or *context*-driven keyword expansions. Web page contents would be pattern-matched against the search terms and a VSM algorithm used to calculate page relevance rankings for comparison using P&R graphs.

For this semantic search comparison experiment, it would be assumed that a number of search context ontologies would be available to guide the “user” in query term selection (keywords/phrases). Provision should also be made for input of terms that may not feature in a specific ontology context but which may be required in the context of the query objective, e.g. the query narrative in the T416 “Three Gorges Project” topic statement (see subsection 3.2.6, Fig. 62) targets documents containing “total cost” and “completion date” information; as these generic terms were not considered solely relevant to a Hydro-electric ontology, the input provision was considered an objective approach, especially as users would likely vary their query term selection during a search process.

Keyword Handling for Ontology-based Query Expansion

Following some preliminary search tool prototyping, inclusive OR and *must have* operators were provided to improve query flexibility.

Search Execution

The basic process for keyword-only search should be straightforward, i.e. the terms would need to be stored and the document corpus systematically scanned for pattern-matches within the text repository. It was envisaged that the process for semantics-based search would require an intermediary stage where, once search context and keyword/query terms had been input, the appropriate ontology context would be loaded and expansion terms identified. The process would then continue as for keyword-only search, but this time pattern-matching the ontology-expanded query terms against the text repository.

Query Expansion Control

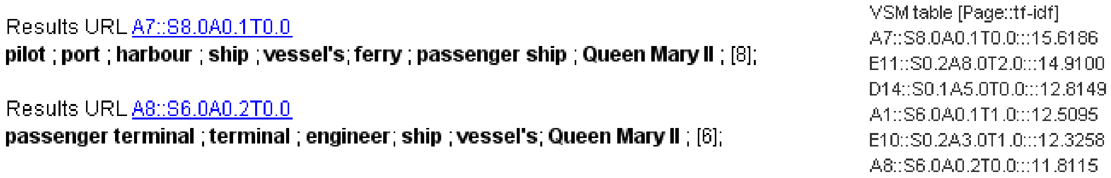
The search process should focus on providing the user with choices between the different query modes, e.g. it was considered that, for *OQE*, SemSeT would require a user to first select a context and then select the required class from a generated class hierarchy. To achieve this, it was decided that SemSeT could exploit ‘adaptive text’ functionality, which would return possible contexts as the user typed in the query subject. Similarly, when a user started to enter a base query term, the context could be interrogated to return concepts matching the leading characters of chosen term. Finally, it was decided that a user should be able to further control ontology query-expansion by selecting options to determine the nature of an expansion, i.e. to return a combination of *sub* and *super* classes of the query term, or simply use the *whole* ontology - see Figs. 35 and 36 (subsection 3.1.4) and Fig. 41 (subsection 3.1.5).

Search Term Pattern Matching and Validation

Search trials were initially conducted using a small document set so that text pattern matches could be manually validated. Any matching issues were resolved by refining the *regular expression* syntax until accurate and correct hits were returned from documents. For example, the regular expression can be used to match either whole words, or words embedded in others, and can handle word variations (e.g. ship, ships, ship's, and –ship) to ensure that term counts are not overstated in the relevance algorithms by treating them incorrectly as different words. When the expression had been refined to generate reliable results the process was tested using a larger controlled document corpus – see TREC in sections 3.3 and 4.1.

Developing a Test Search Corpus

As it was not possible to have access to a search engine document index, SeMSeT's initial search and page ranking tests were conducted by querying a bespoke, online Web corpus (100+ documents). The documents were created with predetermined combinations of relevant and non-relevant terms and were then queried for relevancy against some small test Sea, Air and Tourism ontologies. A code was embedded in each document, to confirm the number of relevant and non-relevant terms based on each of ontologies, e.g. in the code "S3.1A0.2T2.1", "S3.1" denoted 3 relevant and 1 non-relevant Sea terms, "A0.2" no relevant and 2 non-relevant Air terms, etc. For test purposes, search term hits were returned in document order in the main panel and a VSM ranked document relevance list was output in a separate panel, as in Fig. 32.



```
Results URL A7::S8.0A0.1T0.0
pilot ; port ; harbour ; ship ;vessel's; ferry ; passenger ship ; Queen Mary II ; [8];

Results URL A8::S6.0A0.2T0.0
passenger terminal ; terminal ; engineer; ship ;vessel's; Queen Mary II ; [6];
```

```
VSM table [Page::tf-idf]
A7::S8.0A0.1T0.0:::15.6186
E11::S0.2A8.0T2.0:::14.9100
D14::S0.1A5.0T0.0:::12.8149
A1::S6.0A0.1T1.0:::12.5095
E10::S0.2A3.0T1.0:::12.3258
A8::S6.0A0.2T0.0:::11.8115
```

Fig. 32. An extract of typical SemSeT outputs.

Results were validated by manually comparing actual document hits and terms against the predetermined relevance data to confirm the integrity of SemSeT's search results. The controlled corpus also allowed the page relevance calculation and ranking algorithm to be validated.

Schematic of the SemSeT Process

The SemSeT query expansion, search and relevance measurement process is reflected in five key stages shown in Fig. 33, which extends the high-level process shown in Fig. 31; it involves search mode selection (A), base keyword entry (B), ontology traversal for *OQE* candidates and later term weighting, and query term set generation (C), document text analysis using pattern matching and a regular expression (D), term weight allocation for the VSM *tf-idf* document relevance algorithm for P&R (E).

It will be seen that the process only differs in the query term set generation stage, i.e. the *OQE* mode process diverges: to either ontology traversal, to generate the ontology query expansion set, or keyword mode, where the base query terms are forwarded as the query term set. Stage C is therefore determined by the user's search mode selection.

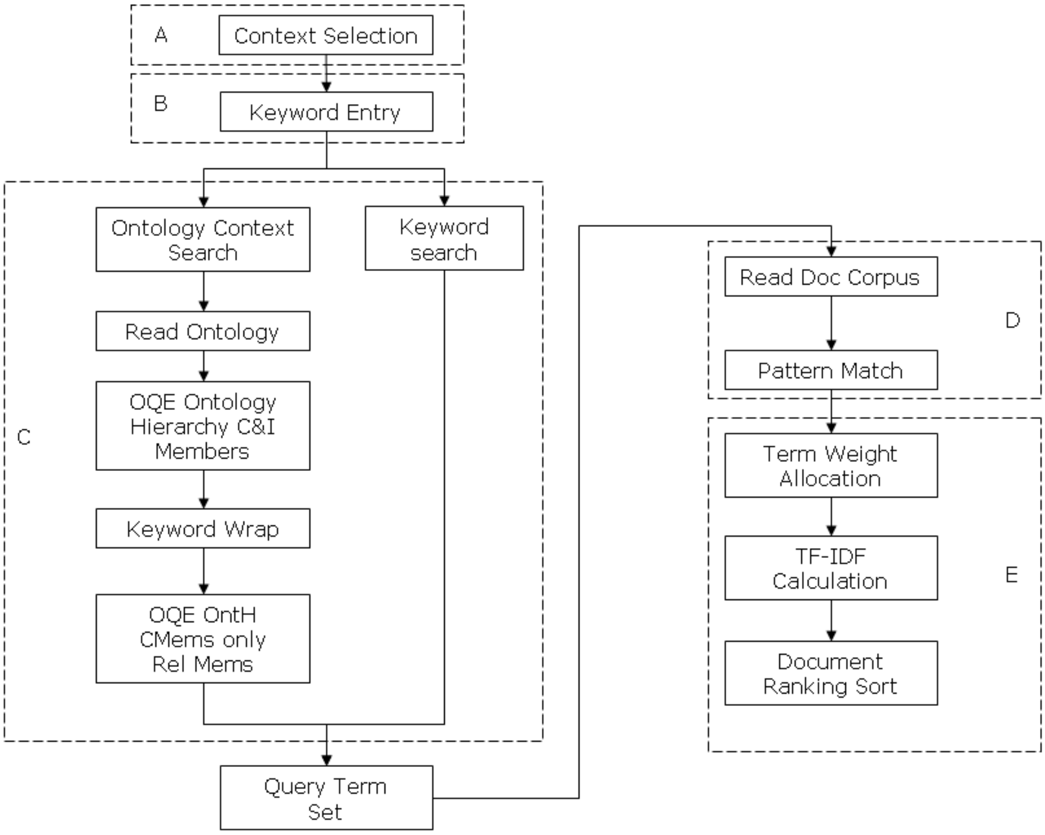
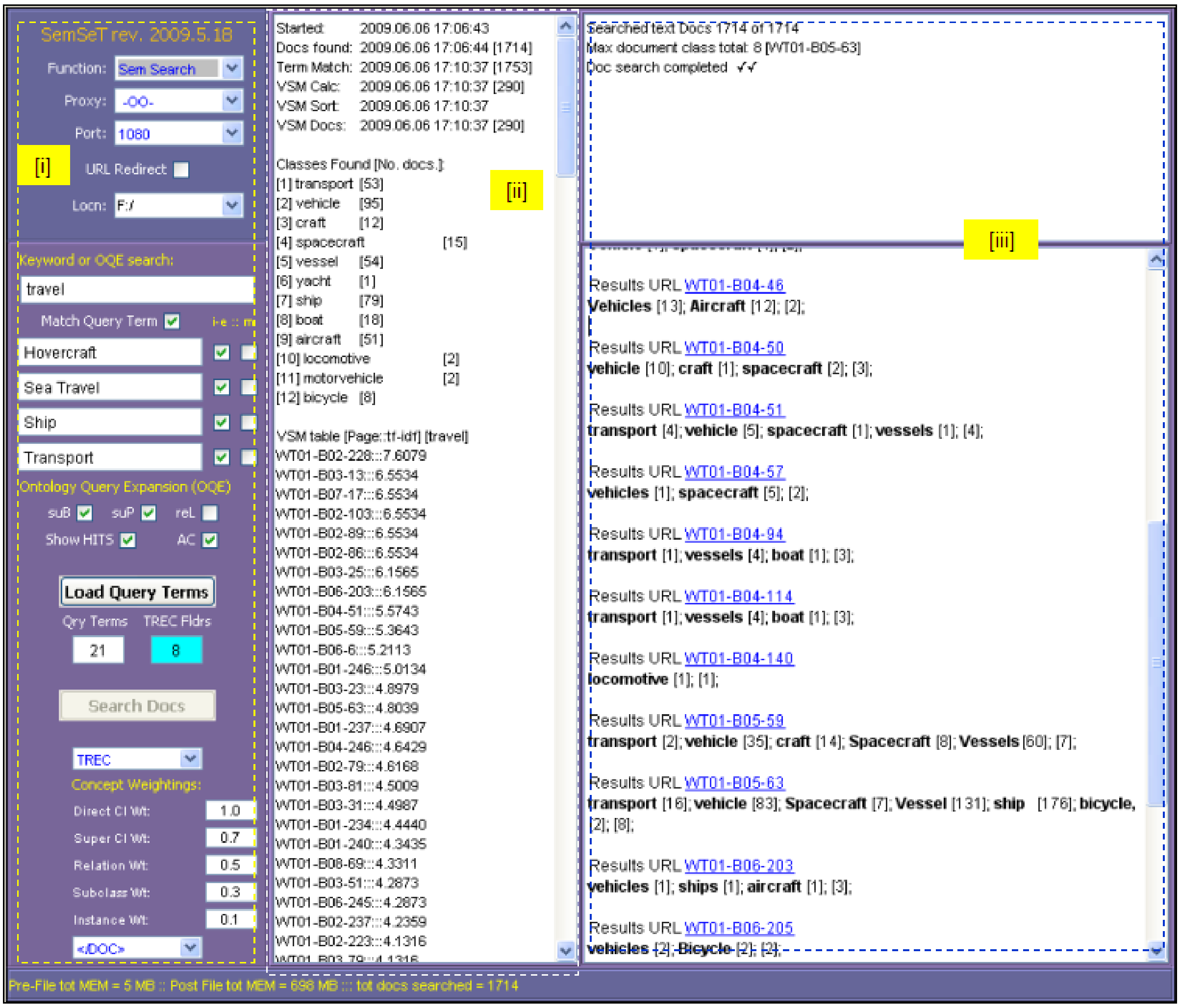


Fig. 33. Key SemSeT search, measurement and comparison process stages.

3.1.3 SemSeT Interface

It was decided that a search tool was required because it was not considered practical or feasible to use commercial search engine platforms, for several reasons: page hits would be dependent on the extent of their own indexing; ontology structures and relevance algorithms could not be incorporated within or at the end of their processes; therefore, meaningful keyword versus ontology comparisons of page relevance would not be possible. The interface is shown in Fig. 34 and has three main components:

- query setup: this involves search context, keyword and query mode selection and is conducted in the panel bounded by the dashed line [i];
- query mode, query term selection and VSM *tf-idf* scoring feedback, based on query setup: this is located in panel [ii], where information is returned to the user relating to search context choices, context class listings, *OQE* term sets for each query term, and resulting document relevance rankings generated by the query;
- query response: output of query term matching results in a ranked document list, shown in panel marked [iii].



SemSeT rev. 2009.5.18

Function: Sem Search
Proxy: -OO-
Port: 1080
URL Redirect □
Locn: F:/

Keyword or OQE search: travel
Match Query Term ☑ i-e :: m
Hovercraft ☑ □
Sea Travel ☑ □
Ship ☑ □
Transport ☑ □

Ontology Query Expansion (OQE)
suB ☑ suP ☑ reL □
Show HITS ☑ AC ☑

Qry Terms: 21 TREC Fldrs: 8

TREC

Concept Weightings:

Direct Cl Wt:	1.0
Super Cl Wt:	0.7
Relation Wt:	0.5
Subclass Wt:	0.3
Instance Wt:	0.1

</DOC>

```
Started:     2009.06.06 17:06:43
Docs found:  2009.06.06 17:06:44 [1714]
Term Match:  2009.06.06 17:10:37 [1753]
VSM Calc:    2009.06.06 17:10:37 [290]
VSM Sort:    2009.06.06 17:10:37
VSM Docs:    2009.06.06 17:10:37 [290]

Classes Found [No. docs.]:
[1] transport  [53]
[2] vehicle    [95]
[3] craft      [12]
[4] spacecraft                [15]
[5] vessel     [54]
[6] yacht      [1]
[7] ship       [79]
[8] boat       [18]
[9] aircraft   [51]
[10] locomotive               [2]
[11] motorvehicle             [2]
[12] bicycle   [8]

VSM table [Page::tf-idf] [travel]
WT01-B02-228:::7.6079
WT01-B03-13:::6.5534
WT01-B07-17:::6.5534
WT01-B02-103:::6.5534
WT01-B02-89:::6.5534
WT01-B02-86:::6.5534
WT01-B03-25:::6.1565
WT01-B06-203:::6.1565
WT01-B04-51:::5.5743
WT01-B05-59:::5.3643
WT01-B06-6:::5.2113
WT01-B01-246:::5.0134
WT01-B03-23:::4.8979
WT01-B05-63:::4.8039
WT01-B01-237:::4.6907
WT01-B04-246:::4.6429
WT01-B02-79:::4.6168
WT01-B03-81:::4.5009
WT01-B03-31:::4.4987
WT01-B01-234:::4.4440
WT01-B01-240:::4.3435
WT01-B08-69:::4.3311
WT01-B03-51:::4.2873
WT01-B06-245:::4.2873
WT01-B02-237:::4.2359
WT01-B02-223:::4.1316
WT01-B03-79:::4.1316
```

```
Searched text Docs 1714 of 1714
Max document class total: 8 [WT01-B05-63]
Doc search completed  ✓✓
```

Results URL WT01-B04-46
Vehicles [13]; **Aircraft** [12]; [2];

Results URL WT01-B04-50
vehicle [10]; **craft** [1]; **spacecraft** [2]; [3];

Results URL WT01-B04-51
transport [4]; **vehicle** [5]; **spacecraft** [1]; **vessels** [1]; [4];

Results URL WT01-B04-57
vehicles [1]; **spacecraft** [5]; [2];

Results URL WT01-B04-94
transport [1]; **vessels** [4]; **boat** [1]; [3];

Results URL WT01-B04-114
transport [1]; **vessels** [4]; **boat** [1]; [3];

Results URL WT01-B04-140
locomotive [1]; [1];

Results URL WT01-B05-59
transport [2]; **vehicle** [35]; **craft** [14]; **Spacecraft** [8]; **Vessels** [60]; [7];

Results URL WT01-B05-63
transport [16]; **vehicle** [83]; **Spacecraft** [7]; **Vessel** [131]; **ship** [176]; **bicycle**, [2]; [8];

Results URL WT01-B06-203
vehicles [1]; **ships** [1]; **aircraft** [1]; [3];

Results URL WT01-B06-205
vehicles [2]; **Bicycle** [2]; [2];

Pre-File tot MEM = 5 MB :: Post File tot MEM = 698 MB ::: tot docs searched = 1714

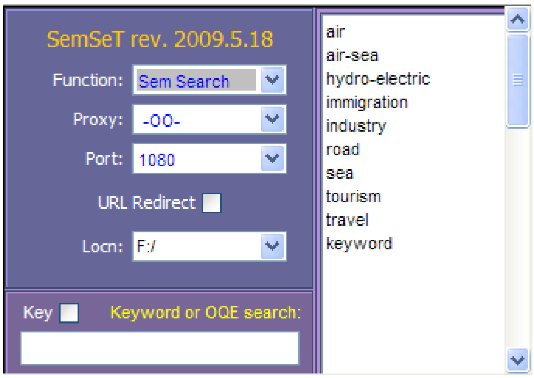
Fig. 34. The SemSeT interface components.

The objective is to guide the user to intuitively populate the search context and query term inputs: panel [i] input boxes use an adaptive text process, i.e. entering or removing input characters in the search mode and query term input boxes generates a list in panel [ii] and populates the input box with a list item, based on the leading characters in the box.

Based on the above, the next subsection discusses in more detail, the key ontology and term selection steps required when inputting and outputting a SemSeT query. A separate analysis, of the SemSeT search process, is presented in the state transition network diagram in Fig. 41, subsection 3.1.5. The query process demonstrated seeks to address some of the user interaction support issues highlighted previously in subsection 1.6.5 (Bhogal et al., 2007).

3.1.4 Making a SemSeT Query

Fig. 35 shows a typical representation of the query setup user interaction between elements [i] and [ii] above. First, if the user clears the “keyword or *OQE* search” box in [i] the adaptive text process will reveal all search mode options in [ii] and entering an initial character reduces the search mode options.



SemSeT rev. 2009.5.18

Function: Sem Search

Proxy: -OO-

Port: 1080

URL Redirect □

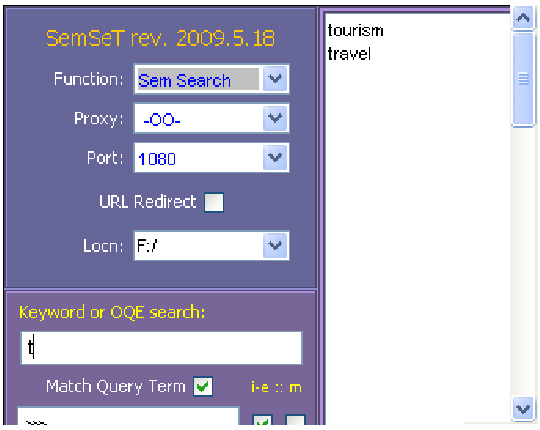
Locn: F:/

Key □ Keyword or OQE search:

air
air-sea
hydro-electric
immigration
industry
road
sea
tourism
travel
keyword

Fig. 35. Displaying all available search modes.

The semantic search mode ontology context choices accessible in the system and based on “t” are shown in Fig. 36, i.e. context modes related to tourism, travel, etc. Further text input further narrows choice until the required context is identified.



SemSeT rev. 2009.5.18

Function: Sem Search

Proxy: -OO-

Port: 1080

URL Redirect □

Locn: F:/

Keyword or OQE search:

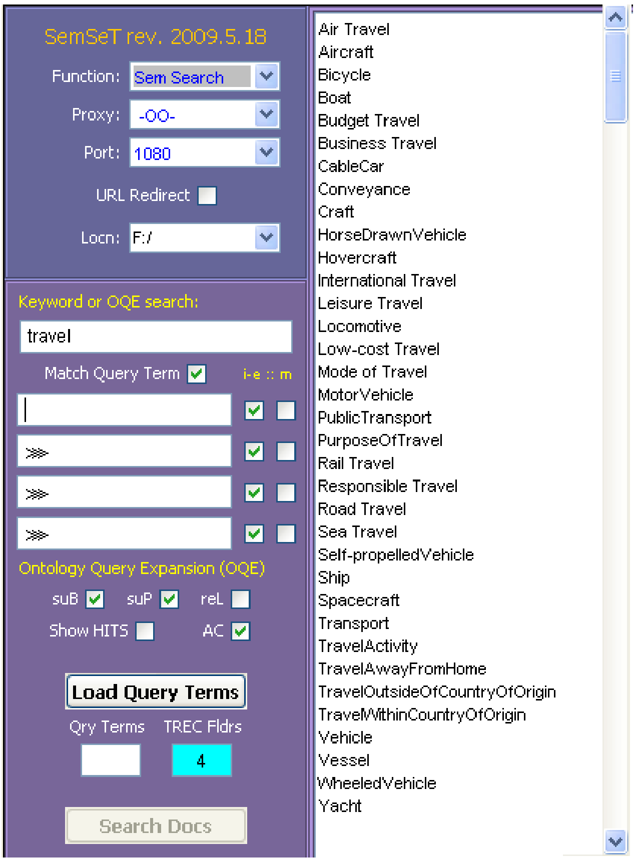
t

Match Query Term ☑ i-e :: m

tourism
travel

Fig. 36. Targeting a search mode for *OQE*.

Assuming the user inputs “tr” the system, SemSeT will interrogate the stored ontology and display all travel classes in [ii] – as shown in Fig. 37.



SemSeT rev. 2009.5.18

Function: Sem Search

Proxy: -OO-

Port: 1080

URL Redirect □

Locn: F:/

Keyword or OQE search:

travel

Match Query Term ☑ i-e :: m

☑ □

≫ ☑ □

≫ ☑ □

≫ ☑ □

Ontology Query Expansion (OQE)

suB ☑ suP ☑ reL □

Show HITS □ AC ☑

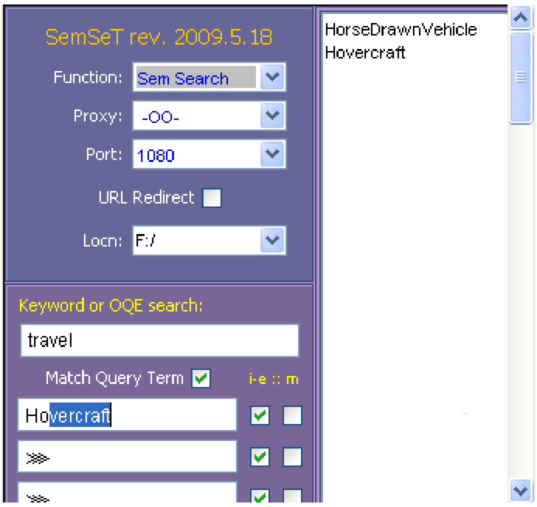
Qry Terms TREC Fldrs

4

Air Travel
Aircraft
Bicycle
Boat
Budget Travel
Business Travel
CableCar
Conveyance
Craft
HorseDrawnVehicle
Hovercraft
International Travel
Leisure Travel
Locomotive
Low-cost Travel
Mode of Travel
MotorVehicle
PublicTransport
PurposeOfTravel
Rail Travel
Responsible Travel
Road Travel
Sea Travel
Self-propelledVehicle
Ship
Spacecraft
Transport
TravelActivity
TravelAwayFromHome
TravelOutsideOfCountryOfOrigin
TravelWithinCountryOfOrigin
Vehicle
Vessel
WheeledVehicle
Yacht

Fig. 37. Candidate query term classes for travel context.

After selecting the search context, the adaptive text input process now requires up to four base query *terms* to drive *OQE*. Fig. 38 shows the leading characters “ho” have generated the only *term* that matches these characters, i.e. concept Hovercraft.



SemSeT rev. 2009.5.18

Function: Sem Search

Proxy: -OO-

Port: 1080

URL Redirect □

Locn: F:/

Keyword or OQE search:

travel

Match Query Term ☑ i-e :: m

Hovercraft ☑ □

≫ ☑ □

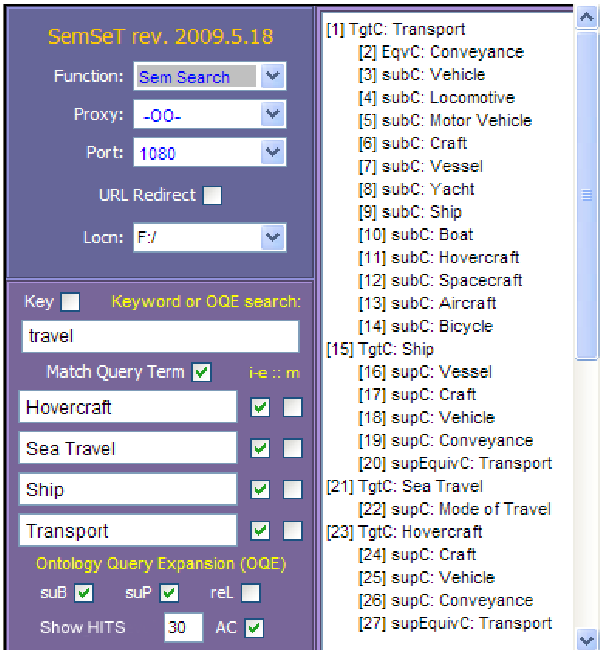
≫ ☑ □

HorseDrawnVehicle
Hovercraft

Fig. 38. Class Hovercraft selected as first query *term* for *OQE*.

Ontology concept selection is repeated for all required query terms, to create a base query term set for the *OQE*; if a matching ontology term is not found against the text input, the user’s input is accepted – as mentioned in subsection 3.1.2. Various *OQE* options can then be chosen, e.g. *sub* and *super* class or *sub* and *super* and *relation* class *OQE*, which are discussed later. Fig. 39

shows the four terms in the base query term set for *OQE*, i.e. Hovercraft, Sea Travel, Ship and Transport. The base query term set is loaded by selecting "Load Query Terms" and the full *OQE* set is listed in feedback panel [ii], i.e. each term with related sub, super and equivalent classes, and any individual terms.



SemSeT rev. 2009.5.18

Function: Sem Search

Proxy: -00-

Port: 1080

URL Redirect □

Locn: F:/

Key □ Keyword or OQE search:

travel

Match Query Term ☑ i-e :: m

Hovercraft ☑ □

Sea Travel ☑ □

Ship ☑ □

Transport ☑ □

Ontology Query Expansion (OQE)

suB ☑ suP ☑ reL □

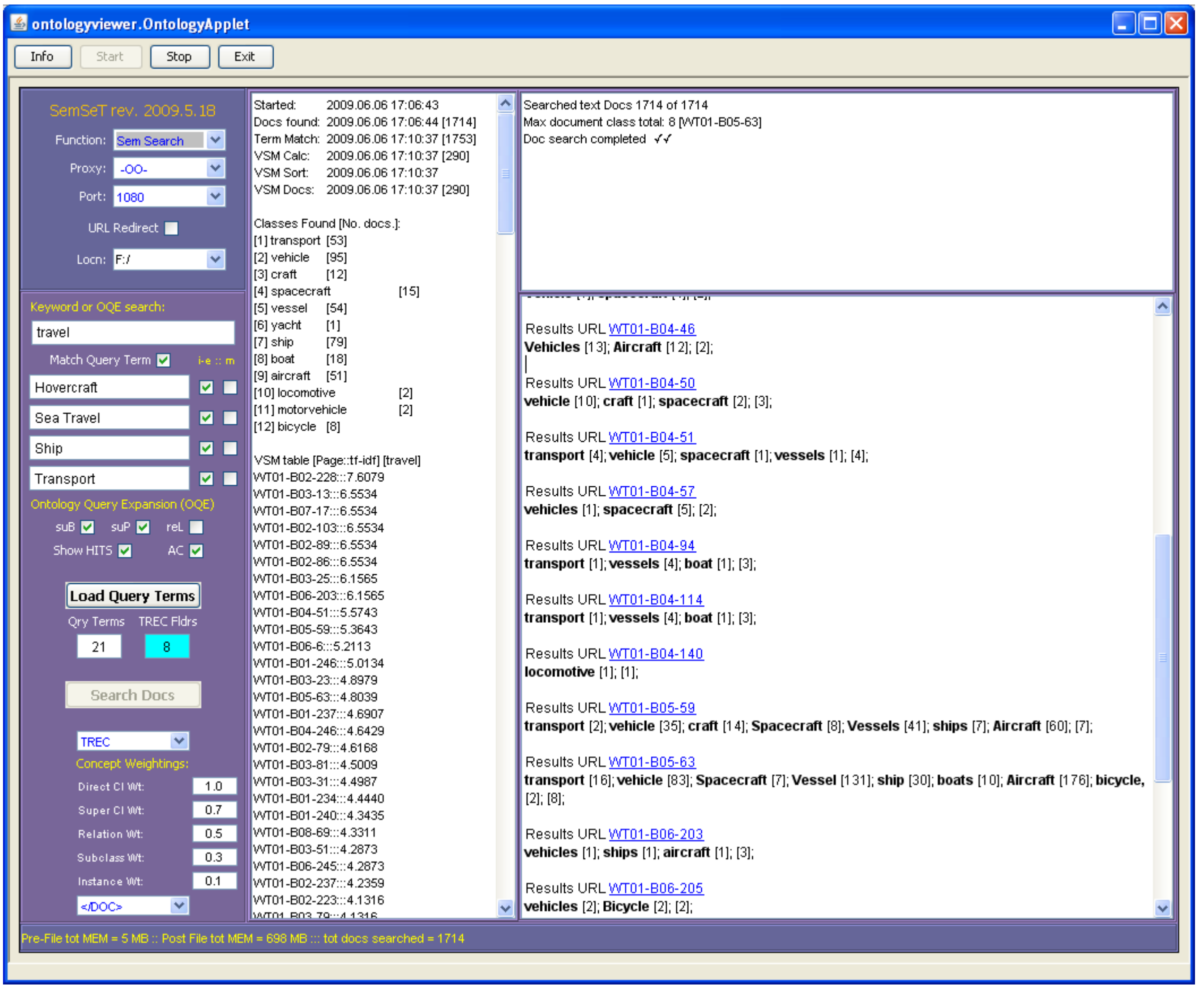
Show HITS 30 AC ☑

```
[1] TgtC: Transport
    [2] EqvC: Conveyance
    [3] subC: Vehicle
    [4] subC: Locomotive
    [5] subC: Motor Vehicle
    [6] subC: Craft
    [7] subC: Vessel
    [8] subC: Yacht
    [9] subC: Ship
    [10] subC: Boat
    [11] subC: Hovercraft
    [12] subC: Spacecraft
    [13] subC: Aircraft
    [14] subC: Bicycle
[15] TgtC: Ship
    [16] supC: Vessel
    [17] supC: Craft
    [18] supC: Vehicle
    [19] supC: Conveyance
    [20] supEquivC: Transport
[21] TgtC: Sea Travel
    [22] supC: Mode of Travel
[23] TgtC: Hovercraft
    [24] supC: Craft
    [25] supC: Vehicle
    [26] supC: Conveyance
    [27] supEquivC: Transport
```

Fig. 39. *OQE* set generated from the base query *terms*.

Depending on the query terms input, the query expansion can generate duplicate expansion terms, e.g. Craft, Vehicle and Vessel; these are automatically filtered prior to document search. Retrieved document and terms, together with relevance scores are then output in ranked order in [ii] and [iii], as shown in Fig. 40. In addition to the document retrieval and relevance information, SemSeT also generates P&R statistics. An example of the P&R data is shown in subsection 3.1.7 (search effectiveness outputs) and Appendix H.

The user interface has been tested during development and in all the experiments and the adaptive text selection, of ontology contexts and terms, functioned reliably and proved effective in helping to guide the "user" in the search process, albeit with a restricted menu of contexts; the assistive algorithms supporting the adaptive text selection process have provided a platform for further research.



Function: Sem Search
Proxy: -OO-
Port: 1080
URL Redirect □
Locn: F:/

Keyword or OQE search: travel
Match Query Term ☑ i-e :: m

	i-e	m
Hovercraft	☑	□
Sea Travel	☑	□
Ship	☑	□
Transport	☑	□

Ontology Query Expansion (OQE)
suB ☑ suP ☑ reL □
Show HITS ☑ AC ☑

Qry Terms: 21
TREC Fldrs: 8

TREC
Concept Weightings:

Direct Cl Wt:	1.0
Super Cl Wt:	0.7
Relation Wt:	0.5
Subclass Wt:	0.3
Instance Wt:	0.1

</DOC>

```
Started:      2009.06.06 17:06:43
Docs found:   2009.06.06 17:06:44 [1714]
Term Match:   2009.06.06 17:10:37 [1753]
VSM Calc:     2009.06.06 17:10:37 [290]
VSM Sort:     2009.06.06 17:10:37
VSM Docs:     2009.06.06 17:10:37 [290]

Classes Found [No. docs.]:
[1] transport  [53]
[2] vehicle    [95]
[3] craft      [12]
[4] spacecraft           [15]
[5] vessel     [54]
[6] yacht      [1]
[7] ship       [79]
[8] boat       [18]
[9] aircraft   [51]
[10] locomotive          [2]
[11] motorvehicle        [2]
[12] bicycle   [8]

VSM table [Page::tf-idf] [travel]
WT01-B02-228:::7.6079
WT01-B03-13:::6.5534
WT01-B07-17:::6.5534
WT01-B02-103:::6.5534
WT01-B02-89:::6.5534
WT01-B02-86:::6.5534
WT01-B03-25:::6.1565
WT01-B06-203:::6.1565
WT01-B04-51:::5.5743
WT01-B05-59:::5.3643
WT01-B06-6:::5.2113
WT01-B01-246:::5.0134
WT01-B03-23:::4.8979
WT01-B05-63:::4.8039
WT01-B01-237:::4.6907
WT01-B04-246:::4.6429
WT01-B02-79:::4.6168
WT01-B03-81:::4.5009
WT01-B03-31:::4.4987
WT01-B01-234:::4.4440
WT01-B01-240:::4.3435
WT01-B08-69:::4.3311
WT01-B03-51:::4.2873
WT01-B06-245:::4.2873
WT01-B02-237:::4.2359
WT01-B02-223:::4.1316
WT01-B03-79:::4.1316
```

```
Searched text Docs 1714 of 1714
Max document class total: 8 [WT01-B05-63]
Doc search completed  ✓✓
```

Results URL WT01-B04-46
Vehicles [13]; **Aircraft** [12]; [2];

Results URL WT01-B04-50
vehicle [10]; **craft** [1]; **spacecraft** [2]; [3];

Results URL WT01-B04-51
transport [4]; **vehicle** [5]; **spacecraft** [1]; **vessels** [1]; [4];

Results URL WT01-B04-57
vehicles [1]; **spacecraft** [5]; [2];

Results URL WT01-B04-94
transport [1]; **vessels** [4]; **boat** [1]; [3];

Results URL WT01-B04-114
transport [1]; **vessels** [4]; **boat** [1]; [3];

Results URL WT01-B04-140
locomotive [1]; [1];

Results URL WT01-B05-59
transport [2]; **vehicle** [35]; **craft** [14]; **Spacecraft** [8]; **Vessels** [41]; **ships** [7]; **Aircraft** [60]; [7];

Results URL WT01-B05-63
transport [16]; **vehicle** [83]; **Spacecraft** [7]; **Vessel** [131]; **ship** [30]; **boats** [10]; **Aircraft** [176]; **bicycle,** [2]; [8];

Results URL WT01-B06-203
vehicles [1]; **ships** [1]; **aircraft** [1]; [3];

Results URL WT01-B06-205
vehicles [2]; **Bicycle** [2]; [2];

Pre-File tot MEM = 5 MB :: Post File tot MEM = 698 MB ::: tot docs searched = 1714

Fig. 40. SemSeT's document and relevance ranking outputs.

3.1.5 User and Search Tool Interaction - State Transitions

SemSeT's user interaction and system functionality can be further explained by referring to the STN diagram shown overleaf in Fig. 41. The diagram displays directed lines that depict a user/system action between connected process states: the narrative above each action line denotes the user's activity and the narrative below the line confirms the system response to achieve the resultant process state. So, both user and system activity will describe the impact of each action. An STN diagram would be particularly relevant for a system developer but is presented here to reflect what could typically be required to make it easier for a user to complete a semantics-based query expansion and search.

The STN has been designed to be self-explanatory but it is perhaps worth clarifying the initial step after starting the process: the *diamond* represents a choice for the user, i.e. the stage when the user has the option of selecting keyword or semantic search by entering characters in the context box – as shown previously in Fig. 36.

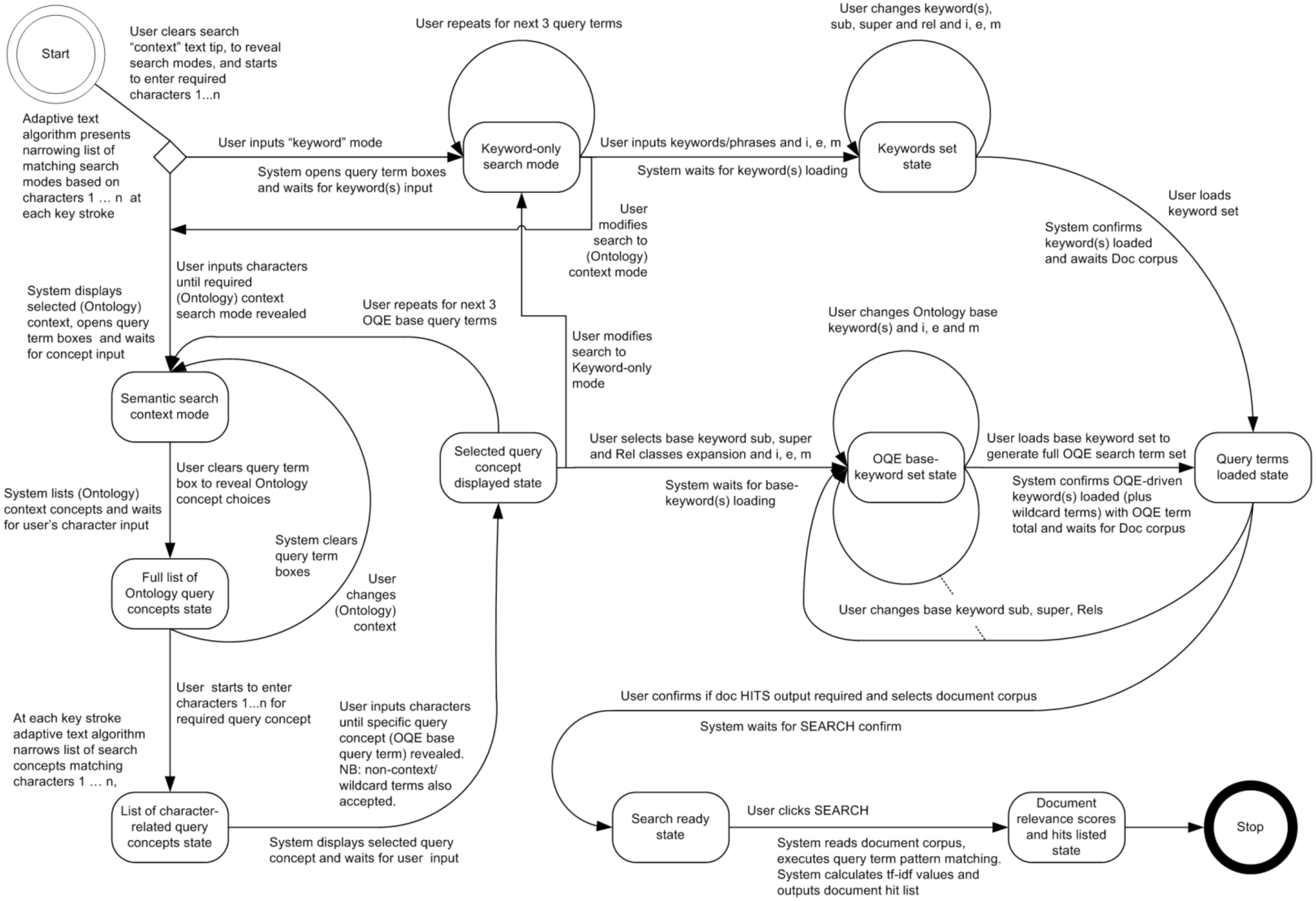

Start
User clears search "context" text tip, to reveal search modes, and starts to enter required characters 1...n
Adaptive text algorithm presents narrowing list of matching search modes based on characters 1 … n at each key stroke
User repeats for next 3 query terms
User inputs "keyword" mode
System opens query term boxes and waits for keyword(s) input
Keyword-only search mode
User changes keyword(s), sub, super and rel and i, e, m
User inputs keywords/phrases and i, e, m
System waits for keyword(s) loading
Keywords set state
User modifies search to (Ontology) context mode
User loads keyword set
System confirms keyword(s) loaded and awaits Doc corpus
User inputs characters until required (Ontology) context search mode revealed
System displays selected (Ontology) context, opens query term boxes and waits for concept input
User repeats for next 3 OQE base query terms
User modifies search to Keyword-only mode
User changes Ontology base keyword(s) and i, e and m
Semantic search context mode
Selected query concept displayed state
User selects base keyword sub, super and Rel classes expansion and i, e, m
System waits for base-keyword(s) loading
OQE base-keyword set state
User loads base keyword set to generate full OQE search term set
System confirms OQE-driven keyword(s) loaded (plus wildcard terms) with OQE term total and waits for Doc corpus
Query terms loaded state
User clears query term box to reveal Ontology concept choices
System lists (Ontology) context concepts and waits for user's character input
System clears query term boxes
User changes (Ontology) context
Full list of Ontology query concepts state
User changes base keyword sub, super, Rels
User starts to enter characters 1...n for required query concept
At each key stroke adaptive text algorithm narrows list of search concepts matching characters 1 … n,
User inputs characters until specific query concept (OQE base query term) revealed. NB: non-context/ wildcard terms also accepted.
User confirms if doc HITS output required and selects document corpus
System waits for SEARCH confirm
List of character-related query concepts state
System displays selected query concept and waits for user input
Search ready state
User clicks SEARCH
System reads document corpus, executes query term pattern matching. System calculates tf-idf values and outputs document hit list
Document relevance scores and hits listed state
Stop

Fig. 41. State Transition Network of imagined query process.

3.1.6 Additional *OQE* Mode Search Options

The keyword to *OQE* comparisons involve between 10 and 20 queries being executed over the document corpus for each TREC query topic. The base keyword query can be manipulated by choosing *OQE* options to extend each keyword/query term based on *all* ontology classes (*All OQE*), *sub and super* classes (*S+S OQE*), or *sub, super and relation* classes (*S+S+R OQE*); these *OQE* options can further be based on various *optional* and *must-have* query term searches:

i. an *optional* query term search: i.e. based on four *optional* keywords (Ko); compared to the keyword-based *S+S OQE optional* term sets (Oo).

ii. a *must-have* query term search: i.e. three *optional* keywords plus one *must-have* keyword (Km); compared to three keyword-based *S+S OQE optional* term sets plus one *must-have* keyword-matching ontology term with related *S+S OQE optional* ontology term set (Om).

The two above query term search options permit *two-way* comparisons, i.e. Ko vs. Oo and Km vs. Om; the next two query term search options permit, incrementally, *three-way* comparisons:

iii. a relation *optional* comparison option (*S+S+R OQE*): i.e. four keyword-based *S+S optional* and relation (*R*) *optional OQE* term sets (Oro); compared to Ko and Oo in (i), i.e. allowing a three-way *optional* comparison of Ko vs. Oo vs. Oro.

iv. a relation *must-have* comparison option (*S+S+R OQE*): i.e. three keyword-based *S+S optional* and relation (*R*) *optional OQE* term sets, plus one *must-have* keyword-matching ontology term with *optional* keyword-based *S+S OQE* and relation (*R*) *optional OQE* term set (Orm); compared to Km and Om in (ii), i.e. allowing a three-way *must-have* comparison of Km vs. Om vs. Orm.

The combinations of *optional* and *must-have* query term search options, for *All*, *S+S* or *S+S+R OQEs* are summarised in *OQE* query mode matrix in Table 2.

Table 2. A matrix of *OQE* options for T401, T416 and T438 queries.

	Ko vs. Oo	Km vs. Om	Ko vs. Oo vs. Oro	Km vs. Om vs. Orm
T401 Immigration	*All OQE, S+S OQE*		n/a	
T416 Three Gorges Project	*S+S OQE*		*S+S+R OQE*	
T438 Tourism	*S+S OQE*		n/a	

The query comparison combinations for *All, S+S, S+S+R OQEs* applied to the Ko, Oo, Oro and Km, Om, Orm query term search options are considered further in section 3.3.

3.1.7 Search Effectiveness Outputs

The TREC corpus included a set of query relevance judgements for each query topic, i.e. listing a pool of relevant and non-relevant documents distributed randomly across the full document collection. The judgement sets allow topic relevant documents to be flagged when calculating P&R search effectiveness measures. The keyword versus *OQE* query outcomes will be based on ranked *tf-idf* document scores, which will allow P&R comparisons to be calculated. Search effectiveness success will be evaluated using P&R graphs containing precision-recall curves for each keyword or *OQE* query executed. Graphs will be presented in the format shown in Fig. 42, i.e. showing scales 0-100% for both precision and recall (unless otherwise stated). For the query mode comparisons outlined in subsection 3.1.6, P&R curve success will be determined by measuring the cumulative number of documents retrieved and the number that are deemed query context relevant; to calculate a cumulative percentage precision at each incremental 10% interval of recall.

Consider the purely hypothetical data shown in Table 3, which assumes there are 50 relevant documents in a document query pool - column (a). In column (c), the first line of *OQE* 2 (10% recall) has resulted in the first 5 relevant documents being retrieved in the top 5 ranked documents returned; therefore, the precision at 10% recall is 100%. At 20% recall (cumulative 10 relevant documents found), the cumulative ranked documents returned were also 10, i.e. 100% precision was achieved at 20% recall. However, 30% recall (cumulative 15 relevant documents), required a total of 16 documents to be returned, resulting in 94% precision at 30% recall. Hypothetically, the most successful query outcome would present a precision-recall curve displaying 100% precision at each recall point; however, search engines can present thousands of *potentially* relevant hits, where relevant documents are often listed over many result pages, such a precision curve would be extremely unlikely in reality.

Table 3. Example of SemSeT P&R data.

	(a)	(b)				(c)	
% R points	Cumulative relevant docs. returned	Cum. docs. returned Keywords only	Keyword % P (a/b)	Cum. docs. returned *OQE* 1	*OQE* 1 % P	Cum. docs. returned *OQE* 2	*OQE* 2 % P
10%	5	6	83%	8	63%	5	100%
20%	10	15	67%	19	53%	10	100%
30%	15	26	58%	32	47%	16	94%
40%	20	39	51%	46	43%	22	91%
50%	25	62	40%	62	40%	31	81%
60%	30	84	36%	79	38%	49	61%
70%	35	112	31%	98	36%	72	49%
80%	40	157	25%	147	27%	111	36%
90%	45	241	19%	214	21%	151	30%
100%	50	356	14%	302	17%	192	26%

As previously highlighted, only a few pages of search engine results are useful, as a typical Web user might only be interested in examining the first page or two of ranked document hits. To recognise this, the TREC query precision results should be considered most indicative in the early, low recall intervals. Therefore, the research experiments will not primarily consider precision outcomes *beyond* 30% recall of primary importance. The success of the TREC *OQE* experiments will be determined by comparing *OQE* P&R curve outcomes against the "base" keyword P&R curve profile. Fig. 42 is based on the data in Table 3 and demonstrates both successful and unsuccessful *OQE* mode outcomes – compared to a keyword query.

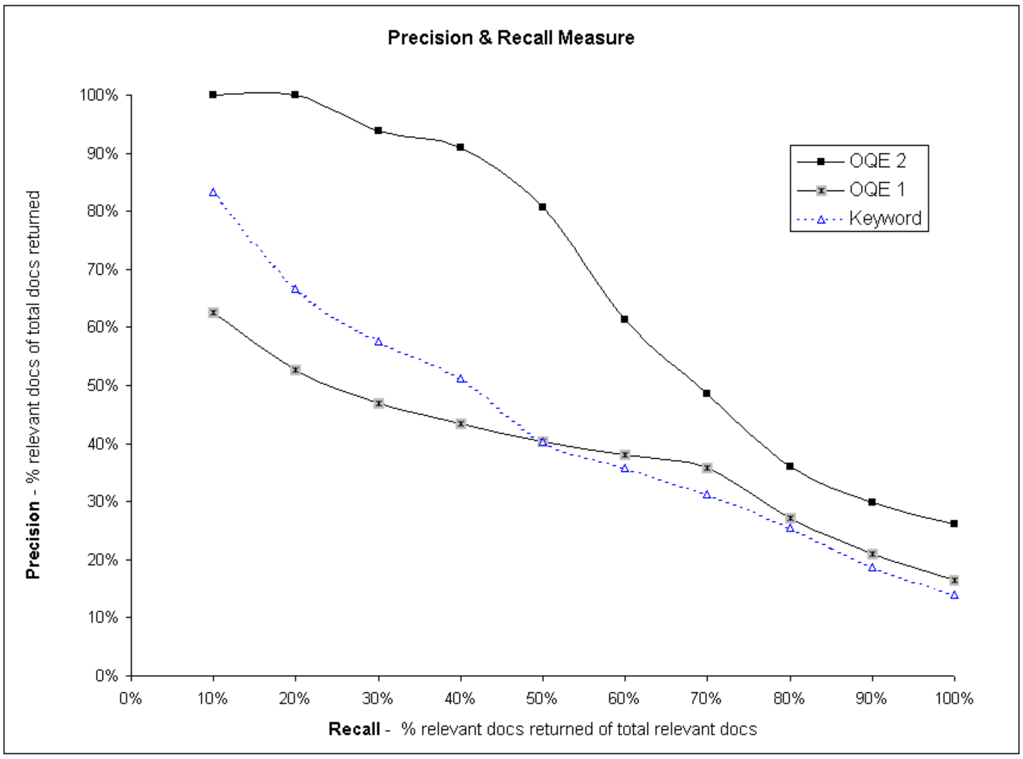


Fig. 42. Graph format for P&R measures.

The keyword P&R curve shows that precision was 83% falling to 58% between 10% and 30% recall respectively. In comparison, *OQE* 1 is considered unsuccessful as it only achieved between 63% and 47% precision (up to 30% recall), despite having higher precision than keyword beyond 50% recall. However, *OQE* 2 has been wholly successful, achieving between 100% and 94% in the same recall range.

Query experiment outcomes, in chapter 4, will show that precision values can fluctuate along the recall axis. To provide a consistent approach in comparing the precision-recall curves, an average of the precision percentage values for the 10%, 20% and 30% recall points (the APV) will be used in performance evaluations. The TREC query experiment precision comparisons will be based primarily on this approach. Table 3's 10%, 20% and 30% recall points provide APV examples.

- Keyword = (83% + 67% + 58%) / 3 = 69% APV.
- *OQE* 1 = (63% + 53% + 47%) / 3 = 54% APV.
- *OQE* 2 = (100% + 100% + 94%) / 3 = 98% APV.

3.2 HOW THE EXPERIMENT WAS DESIGNED

Software programs developed in this research will be described, showing how they facilitated the development of a semantic search tool for *OQE*, ontology contexts and concept weights, search and scoring algorithms, and context ontology design.

3.2.1 Design of SemSeT Interface

Whilst SemSeT provides a prototype search tool to emulate a user's query interface options, it is essentially configured for query experimentation and represents a controlled environment for development and testing of OWL ontology traversal algorithms, to identify query term matching/related *OQE* terms and calculate P&R measures from document relevance scores generated by a modified *tf-idf* algorithm. Therefore, SemSeT is not presented as a fully usable public search tool. Indeed, the tool has no indexing functionality and, as queries are made directly on the TREC document collection, the retrieval experiments are based on a representative document collection cut-off - see comments at the beginning of chapter 4.

3.2.2 Ontology Contexts and *OQE*

The *OQE* process required a set of ontology contexts, i.e. ontologies based on the context of each selected TREC query topic (Foreign Minorities, Three Gorges Project and Tourism). The query topic statement narratives were used to initially conceptualise each ontology context; no prior reference was made to the TREC corpus, e.g. to identify useful concepts. Google was then used to find potentially relevant Web sites to further develop the ontology contexts.

To support keyword-based queries and facilitate document text matching with an ontology class during search, it was decided that wherever possible, classes should be specified at an atomic level (i.e. primitive or self-standing classes) to permit more complex (defined/dependant) classes to be formed by modular construction, e.g. using the best practice concept modularisation approach outlined by Rector (Rector, 2003).

Ontology Specification

Context ontologies were developed firstly using the Protégé ontology editor and then validated with the FaCT++ DL reasoner. Protégé's classification and inferencing process is compatible with FaCT++ and can use it to analyse an ontology hierarchy by identifying any OWL syntax inconsistencies and then correcting and verifying changes to the ontology specification. Ontological consistency is key to ensuring meaningful ontology traversal during the *OQE* process.

Fig. 43 was produced by combining outputs generated by the Protégé graphical tool plug-in (OWLViz) and depicts, in four development stages *a* to *d*, concept specification in a test ontology using Protégé and the use of a classifier to check consistency of the class hierarchy.

As will be seen below, Fig. 43 can illustrate issues that could potentially compromise ontology traversal during *OQE*.

Stage *a* depicts the simple subsumption hierarchy comprising classes D1-D5, each having a clearly defined super class; stage *b* shows how class relationships change when Protégé is used to create two equivalent class relationships, i.e. D5 ≡ D4 and D2 ≡ D3. The corresponding OWL syntax for *a* and *b* are provided in Fig. 44 and show the effect of creating those equivalent class relationships. However, at this stage it can be seen that only equivalent class D2 has an explicit subClassOf relationship, i.e. to D.

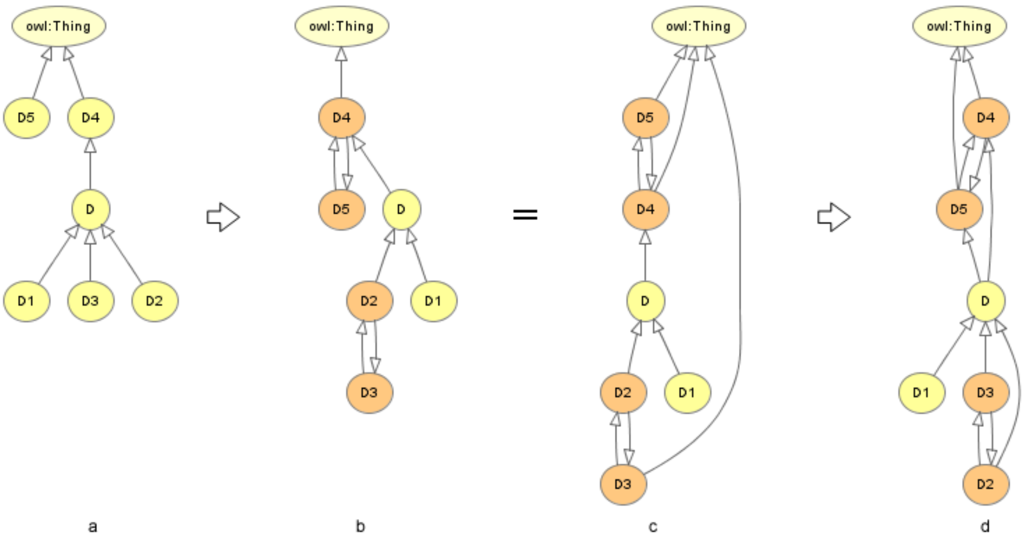


Fig. 43. Stages of test ontology concept specification and classification.

```
<rdf:RDF xmlns=http://www.owl.com/example.owl#>
   <owl:Class rdf:ID="D">
      <rdfs:subClassOf rdf:resource="#D4"/>
   </owl:Class>
   <owl:Class rdf:ID="D1">
      <rdfs:subClassOf rdf:resource="#D"/>
   </owl:Class>
   <owl:Class rdf:ID="D2">
      <rdfs:subClassOf rdf:resource="#D"/>
   </owl:Class>
   <owl:Class rdf:ID="D3">
      <rdfs:subClassOf rdf:resource="#D"/>
   </owl:Class>
   <owl:Class rdf:ID="D4"/>
   <owl:Class rdf:ID="D5"/>
</rdf:RDF>
```

a

```
<rdf:RDF xmlns="http://www.owl.com/example.owl#">
   <owl:Class rdf:ID="D">
      <rdfs:subClassOf rdf:resource="#D4"/>
   </owl:Class>
   <owl:Class rdf:ID="D1">
      <rdfs:subClassOf rdf:resource="#D"/>
   </owl:Class>
   <owl:Class rdf:ID="D2">
      <owl:equivalentClass rdf:resource="#D3"/>
      <rdfs:subClassOf rdf:resource="#D"/>
   </owl:Class>
   <owl:Class rdf:ID="D3">
      <owl:equivalentClass rdf:resource="#D2"/>
   </owl:Class>
   <owl:Class rdf:ID="D4">
      <owl:equivalentClass rdf:resource="#D5"/>
   </owl:Class>
   <owl:Class rdf:ID="D5">
      <owl:equivalentClass rdf:resource="#D4"/>
   </owl:Class>
</rdf:RDF>
```

b

Fig. 44. OWL syntax at specification stages *a* and *b*.

However, when the OWL file was saved and then subsequently re-opened, stage c, it was found that any class *not* explicitly having a super class specified (see Fig. 44, syntax b) was classified as a sub class of the *root* node even though the class may be described as an equivalent class, e.g. as shown in stage c, the Protégé OWLViz plug-in presented both D3 and D5 as subClassOf the root node. Whilst this stage is logically acceptable the hierarchy returned in stage c is incomplete and can be more accurately classified and rationalised, because D3 and D are implicitly subClassOf D and D5 respectively through both an equivalence and transitive relationship, i.e. equivalence between two classes describes their recursive subClassOf relationship and further, if D2 ≡ D3, and D2 subClassOf D, then D3 subClassOf D; similarly, if D subClassOf D4, and D5 ≡ D4, then D subClassOf D5.

Following some SemSeT search experiments, it was established that, given Protégé and SemSeT both use Jena toolkit libraries, the SemSeT ontology traversal process also interpreted the hierarchy in the same way as Protégé OWLViz, i.e. at stage c a SemSeT search for *sub classes* on D did not identify D3; as it was described in the file syntax solely as an equivalent class. This search ontology traversal problem was resolved by classifying the ontology using the Protégé/FaCT++ tool. The result is shown in stage d, where inferencing has classified the full relationships between D3 and D, and D with D5, then modified the OWL file.

It should be pointed out that the process discussed above was carried out while using Protégé version 3.3.1. Subsequent improvements to Protégé, in version 3.4.1, automatically update the file to reflect stages b and c, whilst the file is being developed in the editor; however, a reasoning tool is still required to complete the classification stage d.

Ontology Traversal Example

The above issues are considered in the more detailed example ontology in Fig. 45. The ontology contains various concepts arranged in trees (e.g. A, A1-A4, T, Q etc. – the letters have no particular meaning themselves), with four concepts A, B, C and D, each representing a keyword-matching class (KMC) with potential sub, super or equivalent classes. Certain classes also have asserted conditions, in this case where an object property ac_hasFriend has been used to specify defined relationships; these are represented by dotted lines, i.e. D4 ac_hasFriend Q and N, D1 ac_hasFriend C , A4 ac_hasFriend C6 , and B3 ac_hasFriend D2.

As in the previous subsection, the ontology has been classified to ensure full subsumption consistency regarding transitive and equivalent class relationships. The ontology will be used to demonstrate the ontology traversal paths executed in SemSeT's *OQE* process.

The SemSeT *OQE* process creates a query term (QT) set for each KMC, i.e. it expands the keyword (or phrase, e.g. hydro-electric dam) by traversing the ontology class hierarchy and adding sub, super and relation classes to the QT set, based on the ontology search mode option required.

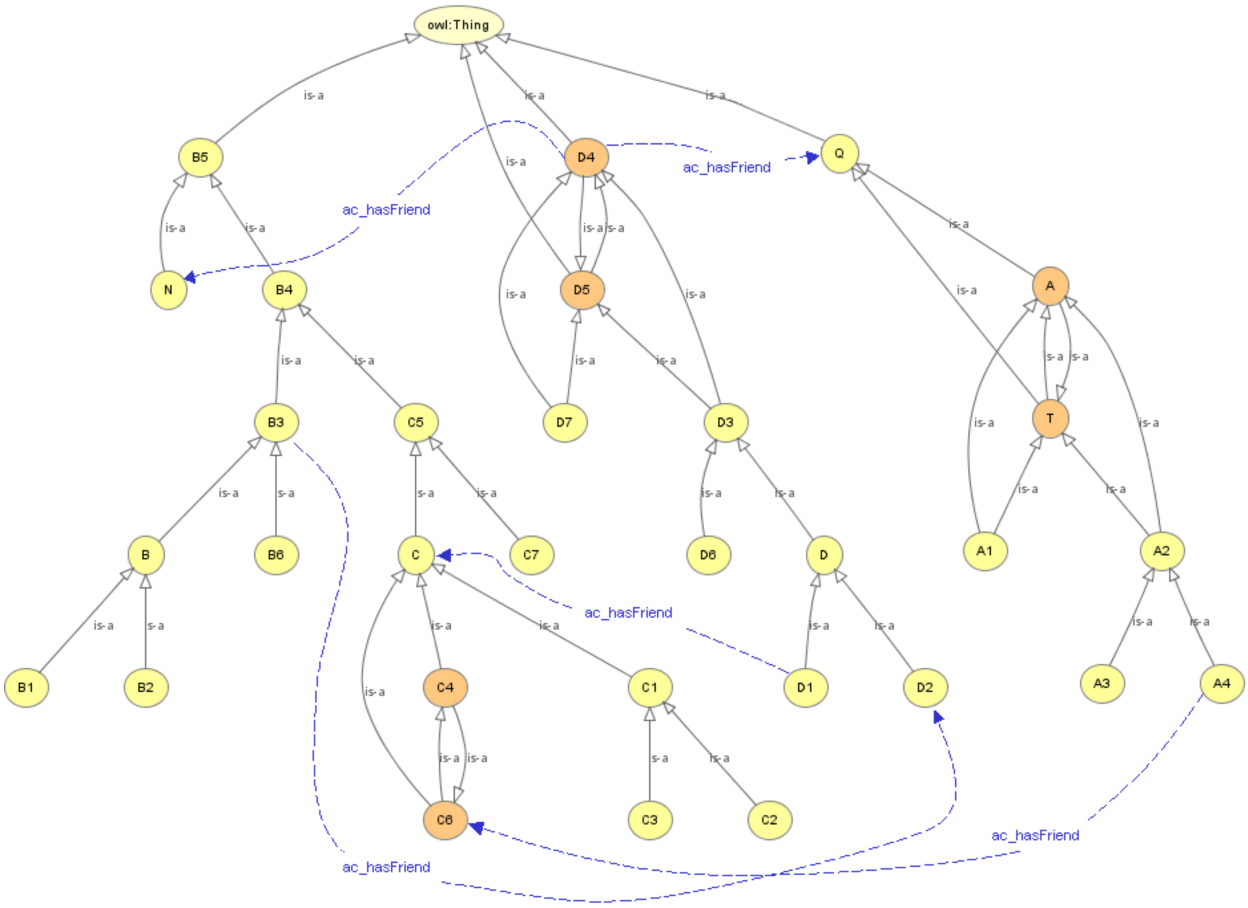


Fig. 45. Ontology relationships for concepts A, B, C and D.

The QT set options referred to in the experimentation chapter are (i) the QT and related sub classes (*S OQE*), (ii) the QT plus sub and super classes (*S+S OQE*), or (iii) QT plus *S+S* with relation classes *R* (*S+S+R OQE*). These options can be summarised as:

(i). *S OQE* implies classes explicitly specified as sub classes of each KMC.

(ii). *S+S OQE* implies (i) plus all "direct" super classes of each KMC, i.e. it excludes any direct super class sub tree, e.g. in Fig. 45, direct super classes of class B would be B3, B4 and B5 only; B6, N and C5 plus its sub classes are ignored.

(iii). *S+S+R OQE* implies (ii) plus any relation classes (i.e. defined by *asserted conditions*) for *every* class identified by *S+S OQE*. Only classes identified by *S+S OQE* are traversed to identify *R* classes.

A further option can be to simply select all ontology classes (*All OQE*) for the query expansion.

The above search mode ontology traversal options will now be used to demonstrate the different *OQE* outcomes using the four KMCs A, B, C and D, in Fig. 45. Firstly, Table 4 shows the expansion outcomes for query term matches on A and C, using ontology traversal *S OQE* mode. The *OQE* has created a set of 12 classes: with class A harvesting equivalent class T and sub classes A1-A4; and target class C generating sub classes C1-3, C4, and C6.

Table 4. *S OQE* traversal outcomes.

A plus (*S*)	**C plus (*S*)**
TgtC: A	TgtC: C
EqvC: T	subC: C6
subC: A1	subC: C4
subC: A2	subC: C1
subC: A4	subC: C2
subC: A3	subC: C3

Next, using *S+S OQE* mode, query term matches with B, C and D reveal the ontology traversal outcomes shown in Table 5, where target class B expands to include super classes B3-5 then sub classes B1-2; similarly, class C expands to super classes C5 plus B4, B5 (again) then sub classes C1-3, C4, C6. Class D expands to include super classes D3-5 then sub classes D1-2.

Table 5. *S+S OQE* traversal outcomes.

B plus (*S+S*)	**C plus (*S+S*)**	**D plus (*S+S*)**
TgtC: B	TgtC: C	TgtC: D
superC: B3	superC: C5	superC: D3
superC: B4	superC: B4	superC: D4
superC: B5	superC: B5	superC: D5
subC: B2	subC: C6	subC: D1
subC: B1	subC: C4	subC: D2
	subC: C1	
	subC: C2	
	subC: C3	

The traversal generates an initial *OQE* set of 21 classes. B4 and B5 are then removed in a duplicated class filtering stage, resulting in 19 classes in the *OQE* set.

Finally, using *S+S+R OQE* mode, query term matches on A, B and D will result in the traversal outcomes shown in Tables 6 and 7. The *S+S+R OQE* result is generated in two stages. In the first stage, class A expands the *S OQE* mode result, shown in Table 4, to also include super class Q, whereas B and D return the same results shown in Table 5; consequently, the first stage results in the query expansion having 19 classes in total and no duplicates.

Table 6. *S+S OQE* traversal outcomes.

A plus (*S+S*)	**B plus (*S+S*)**	**D plus (*S+S*)**
TgtC: A	TgtC: B	TgtC: D
EqvC: T	superC: B3	superC: D3
superC: Q	superC: B4	superC: D4
subC: A1	superC: B5	superC: D5
subC: A2	subC: B2	subC: D1
subC: A3	subC: B1	subC: D2
subC: A4		

In the second stage, the algorithm takes A, B and D's complete 19 class set and searches for asserted condition (relation) classes that were specified using the hasFriend relation; this results in 5 classes D2, C, Q, N and C6 being identified (of which C6, C and N are new), with D2, Q and N found several times through inheritance, e.g. B1, B2 and B3 inherit the asserted condition from B – see Table 7. The 3 additional relation classes increase the A, B and D *OQE* set to 22.

Table 7. *S+S+R OQE* traversal outcomes.

Additional asserted condition (relation) classes: D2, C, Q, N, C6 via:		
A plus (*S+S+R*)	**B plus (*S+S+R*)**	**D plus (*S+S+R*)**
A4 hasFriend C6 (direct)	B3 hasFriend D2 (direct)	D4 hasFriend Q, N (direct)
	B hasFriend *D2 (inherited)*	D1 hasFriend C (direct), *Q, N (inherited)*
	B1 hasFriend *D2 (inherited)*	D hasFriend *Q, N (inherited)*
	B2 hasFriend *D2 (inherited)*	D3 hasFriend *Q, N (inherited)*
		D2 hasFriend *Q, N (inherited)*
NB: classes in *italics* are identified through inheritance.		

It can be seen from the above that, for any query matched term, *S+S OQE* and *S OQE* modes would be subsumed by a choice of *S+S+R OQE*.

The full extent of traversals, assuming A, B and D *S+S+R* and C *S+S*, is shown in Fig. 46. It should be noted that those classes denoted with dual colours represent classes identified by more than one query term traversal, whilst classes B6, C7, D6 and D7 were not identified at all as they did not represent KMC direct super classes. A duplicate class filter would result in the combined traversals generating an *OQE* set of 22 classes. The inclusion of filters after the *S*, *S+S* and *S+S+R OQE* stages are useful for limiting document search iterations and, more importantly ensure that SemSeT's subsequent scoring algorithm will not duplicate term scores and inflate document relevance scores.

The above traversal examples show how SemSeT's *OQE* algorithms would have identified sub, super, equivalent and relation classes. However, even though ontology classification has resulted in a consistent ontology specification for the *OQE* process to correctly identify *S*, *S+S* and *S+S+R OQE*s, the results conceal several situations that are not immediately obvious.

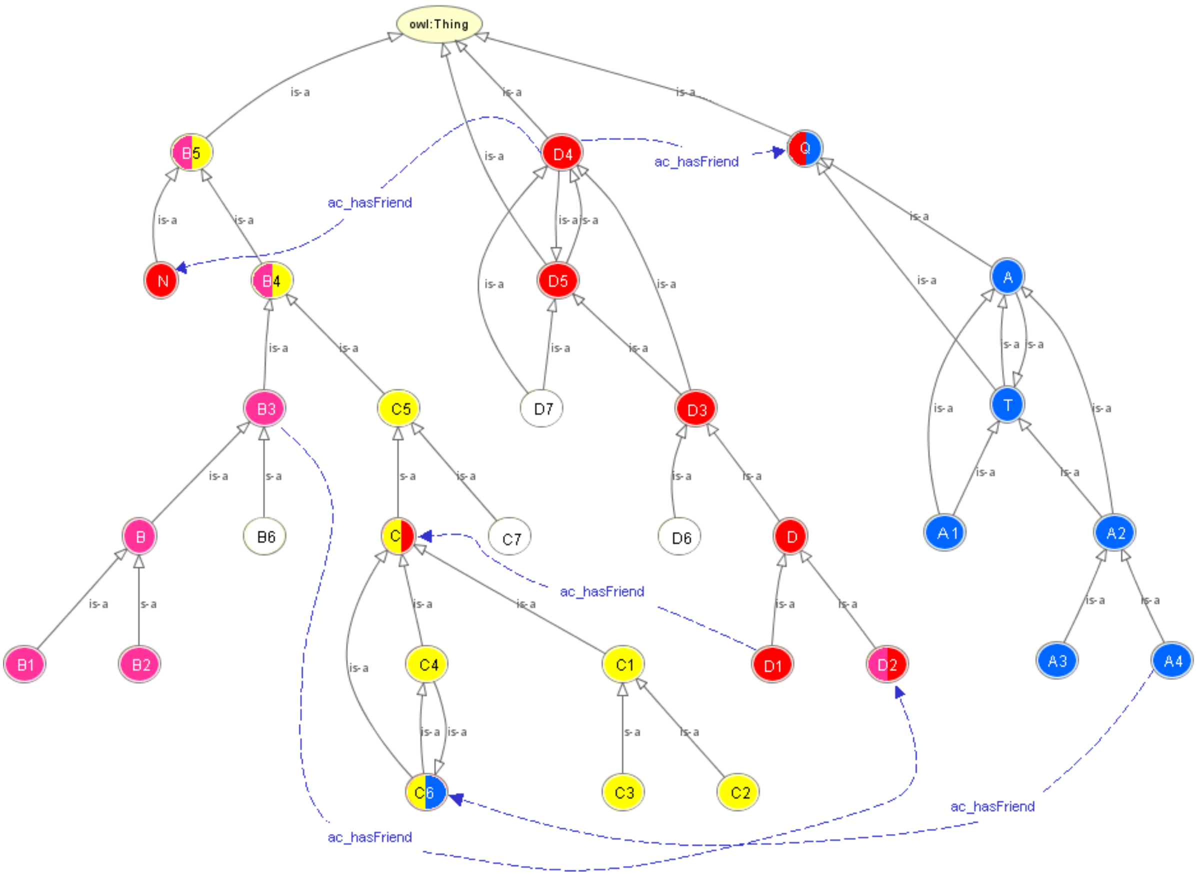


Fig. 46. Extent of ontology traversal for concepts A, B, C and D.

An initial test query on A, based on *S+S OQE* mode, did not in fact return T as the ontology does not explicitly describe T as a sub class of A - see OWL syntax in Fig. 47.

```
…
<owl:Class rdf:ID="T">
      <owl:equivalentClass rdf:resource="#A"/>
      <rdfs:subClassOf rdf:resource="#Q"/>
   </owl:Class>
…
```

Fig. 47. OWL syntax for class T.

Had the query term been based on Q then T would have been found, as a sub class of Q. The resultant *OQE* set was in fact only possible because the inheritance hierarchy class algorithm (see subsection 3.2.4, Fig. 58) was later modified to additionally search for specific equivalent classes. Similarly, a query on either of D4, D5, C4 or C6 would have relied on the same capability to find their respective equivalent class. This situation only occurs *between* equivalent classes.

During testing it was also found that *S+S+R OQE* mode involving A4 would determine the hasFriend relation with C6 but would not identify C6's equivalent class relationship with C4; in fact this was not achieved by running the classifier after the asserted condition relation class had been created. A search for equivalent class has subsequently been included in the relation class algorithm (see subsection 3.2.4, Fig. 59) to ensure inclusion in the *OQE* set.

On reflection, the requirement to run a classifier during ontology development might have been unnecessary if it had been decided to use the Jena Ontology API inferencing capability in SemSeT, i.e. to find inferred relationships.

Super Class Propagation

The semantic search objective is to identify contextually relevant *OQE*-based terms to increase document weighting and recognise the degree of relevance of returned document. The ontology traversal of super classes shown in the previous subsection, i.e. a generalisation traversal, could have been executed in two ways:

i. include only KMC super classes in *OQE*, i.e. all direct generalisation classes but *not* their sub class branches (see (ii) "direct" classes in previous traversal example);

ii. include all direct KMC super classes in *OQE and* their associated sub class branches; note, including root node Thing here would have resulted in all ontology concepts being added to the *OQE*, in effect an *All OQE*.

Both traversals assume that the top-most super class is a named class, i.e. not the root node Thing, and for this subsection it is assumed all KMC sub classes (specialisation traversal) would be retrieved, by default, in both traversals.

The goal of collecting a KMC sub class (specialised) set is to improve precision and/or recall without introducing heterogeneity. Incrementally, by having concept propagation that includes only identified KMC direct super classes, the risk of heterogeneity may be reduced and precision and/or recall increased although KMC super class (generalised) propagations could affect precision. Similarly, wider concept propagation, involving all sub class branches of identified KMC super classes, may mean that concept heterogeneity is increased and precision/recall decreased.

Fig. 46 shows that a super class traversal from the KMC only extends the *OQE* by adding classes found in the direct super class line (i.e. traversal (i) above), e.g. a keyword match with target class C expands the *OQE* set by including only super classes C5, B5 and B4 – see Table 5; in other words, sub classes of any super classes are ignored in the generalisation traversal. Given that a user's query term is assumed to denote the search context, a user might only retain contextual search control (return concepts homogeneous to the query context) when using traversal (i); whereas traversal (ii) could compromise contextual control by returning potentially heterogeneous concepts, as will be demonstrated next, using the ontology structure shown in Fig. 48. Finally, sibling classes of any KMC class (i.e. other sub classes of the KMC's immediate super class) cannot be assumed to be necessarily homogeneous to the context of the KMC.

A short query experiment was conducted to compare traversal (i) and (ii) outcomes, using a small test Land-Sea-Air ontology containing a fragment of imported SUMO concepts – as shown in Fig. 48.

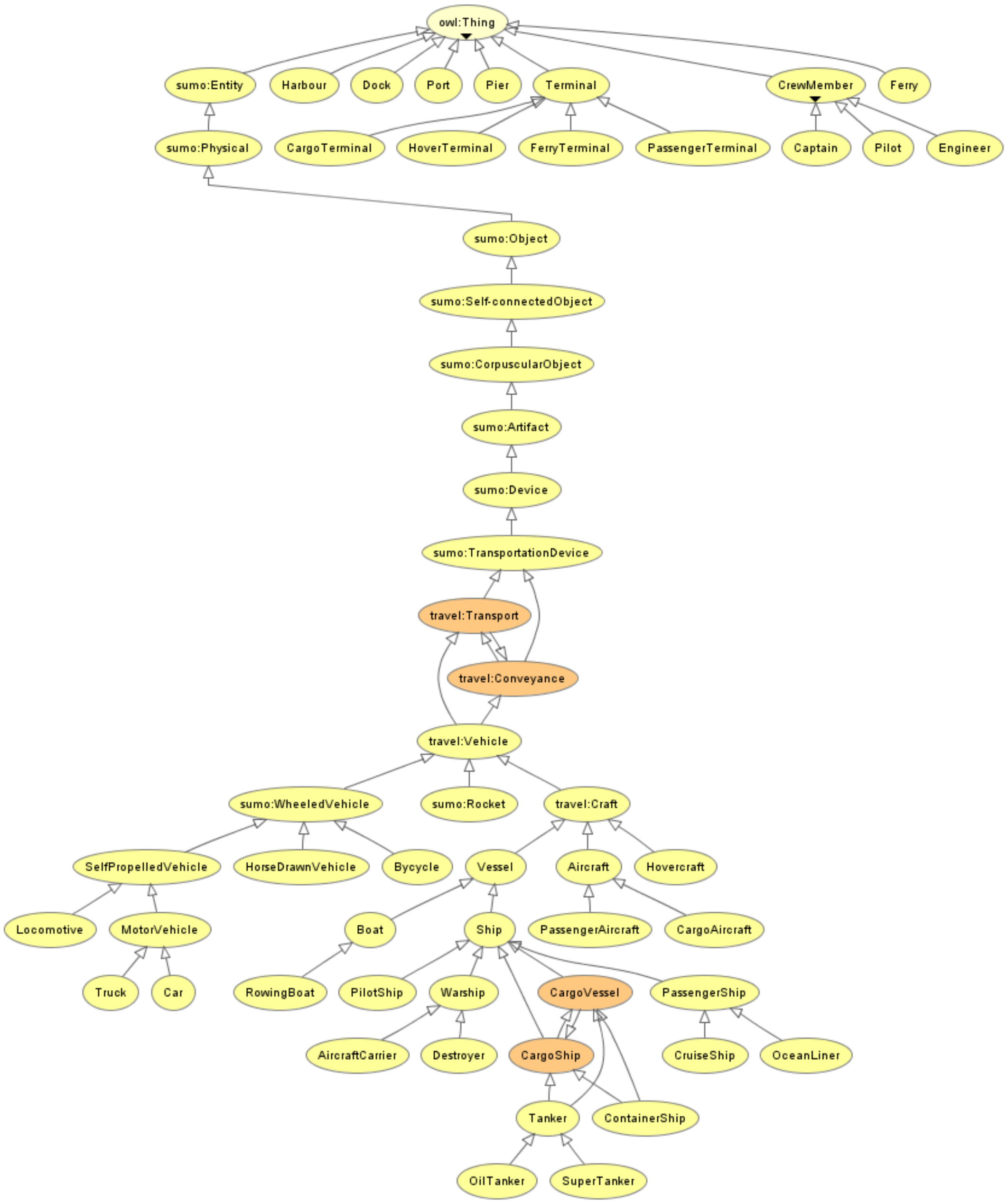


Fig. 48. Land-Sea-Air ontology used to compare traversal (i) and (ii) propagations.

If traversal (ii) above were adopted in the Land-Sea-Air ontology, how search effective would *OQE* propagation be if, e.g. a query on CargoShip returned Ship, then PassengerShip and Warship, together with their respective sub classes? Similarly, returning Ship and Vessel, would also return Boat and RowingBoat and, taking (ii) a stage further, a propagation Ship, Vessel, Craft and Vehicle returns completely heterogeneous concepts, e.g. Aircraft, Rocket, Car, Bicycle etc.

For the test, two query term sets were used: the first contained CargoShip, CargoTerminal, Port and Captain; the second comprised PassengerAircraft, PassengerTerminal, Pier and Captain.

Three queries were made for each, comparing keyword-only to *S+S OQE* using traversal (i) (direct super class line) and then using traversal (ii) (all super class tree). It is clear from the P&R outcomes shown in Fig. 49 and Fig. 50 that traversal (i) produced the best results in both query sets, with traversal (ii) being outperformed by keyword-only up to the 50% recall point.

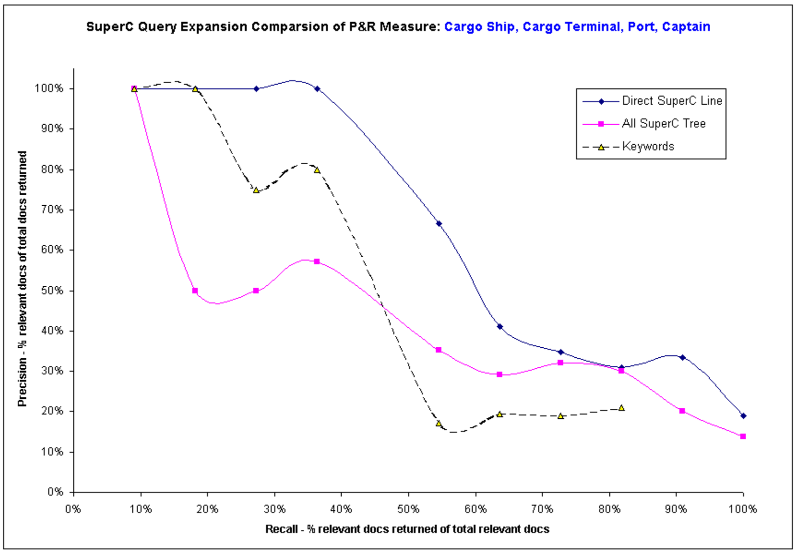


Fig. 49. P&R results using Sea concepts.

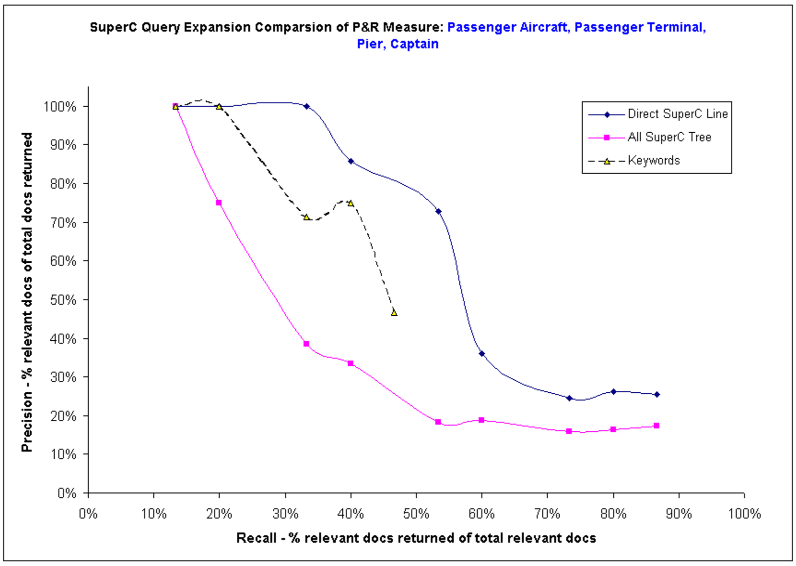
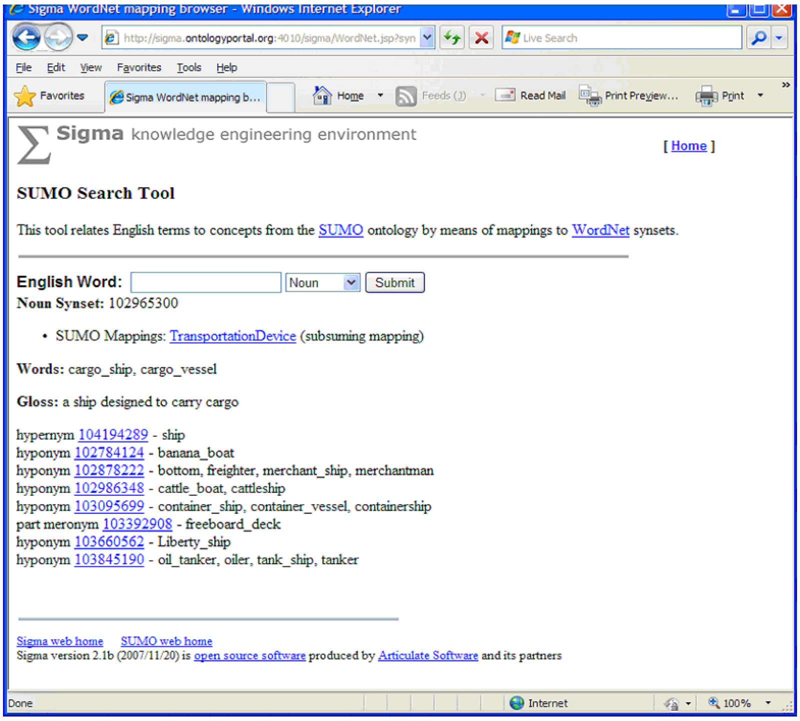


Fig. 50. P&R results using Air concepts.

On the basis of these results, it would appear that any query put in context, e.g. by specifying the term CargoShip, should provide a contextualised *OQE* by ascending direct super classes and harvesting the full sub class tree; not surprisingly, SUMO adopts a similar convention to traversal (i), as can be found in the SUMO Sigma portal [1] and as shown in the SUMO response for the term cargo_ship in Fig. 51. The choice of traversal (i) was ultimately considered justified because P&R measures could be compromised; based on the outcome of document *tf-idf* relevance scores resulting from a wider *OQE* algorithm.

[1] Sigma: http://sigma.ontologyportal.org:4010/sigma/WordNet.jsp?synset=102965300.



Sigma WordNet mapping browser - Windows Internet Explorer
http://sigma.ontologyportal.org:4010/sigma/WordNet.jsp?syn
Live Search
File Edit View Favorites Tools Help
Favorites
Sigma WordNet mapping b...
Home
Feeds (J)
Read Mail
Print Preview...
Print
Sigma knowledge engineering environment
[Home]
Noun
Submit
Done
Internet
100%

SUMO Search Tool

This tool relates English terms to concepts from the SUMO ontology by means of mappings to WordNet synsets.

English Word:

Noun Synset: 102965300

- SUMO Mappings: TransportationDevice (subsuming mapping)

Words: cargo_ship, cargo_vessel

Gloss: a ship designed to carry cargo

hypernym 104194289 - ship
hyponym 102784124 - banana_boat
hyponym 102878222 - bottom, freighter, merchant_ship, merchantman
hyponym 102986348 - cattle_boat, cattleship
hyponym 103095699 - container_ship, container_vessel, containership
part meronym 103392908 - freeboard_deck
hyponym 103660562 - Liberty_ship
hyponym 103845190 - oil_tanker, oiler, tank_ship, tanker

Sigma web home   SUMO web home
Sigma version 2.1b (2007/11/20) is open source software produced by Articulate Software and its partners

Fig. 51. SUMO query response format.

Complex Class Specification

During development of SemSet's *OQE* algorithms, the capability was provided to identify and extract component classes within "complex" *defined* classes.

OWL classes are described on the basis of their super classes, i.e. either named classes or restrictions referred to as anonymous classes. Super classes can also be created using what are termed "complex descriptions"; in constructs that combine named classes using logical operators, e.g. an intersection class of named classes using the AND (⊓) operator. An example of an intersection class is illustrated in Fig. 52, where an anonymous class is used to specify another class description, e.g. we might need to describe class type Father as being equivalent to the intersection of Male and Parent or, more formally Father ≡ (Male ⊓ Parent).

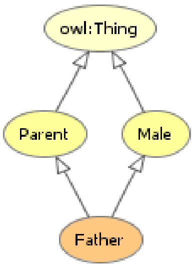


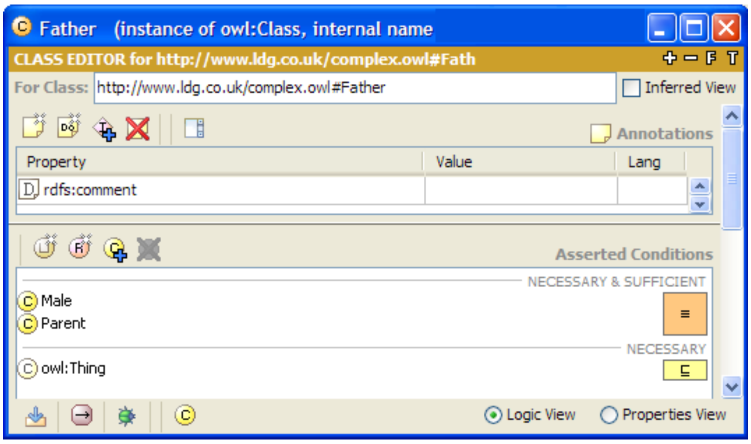
Fig. 52. Visualisation of an intersection class.

However, as the graph does not fully reveal the relationship, Fig. 53 shows the syntax of the anonymous owl:class containing the individual classes.

```
…
<owl:Class rdf:ID="Male"/>
<owl:Class rdf:ID="Father">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Male"/>
        <owl:Class rdf:about="#Parent"/>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:ID="Parent"/>
…
```

Fig. 53. The syntax of an anonymous class containing individual classes.

The syntax describes the relationships in a chained triple, i.e. Father equivalent_to owl:Class, owl:Class intersection_of (Male and Parent). The construct is represented in the Protégé class editor, as shown in Fig. 54.



Father (instance of owl:Class, internal name

CLASS EDITOR for http://www.ldg.co.uk/complex.owl#Fath

For Class: http://www.ldg.co.uk/complex.owl#Father □ Inferred View

Property	Value	Lang
rdfs:comment		

Asserted Conditions

NECESSARY & SUFFICIENT: Male, Parent ≡

NECESSARY: owl:Thing ⊑

● Logic View ○ Properties View

Fig. 54. An anonymous class describing an equivalent class in Protégé.

Whenever the *OQE* algorithm encounters a complex class, Jena Ontology API methods are required to break out the complex class members, i.e. by listing the operands. The Java code in Appendix B illustrates the methods applied; a summarised *OQE* procedure follows and an extended algorithm is shown in subsection 3.2.4, Fig. 58.

3.2.3 Design of Ontology Traversal and Scoring Algorithms

To provide a preliminary high-level understanding of key algorithms, pseudo code of the Java program iterations are shown for the *inheritance class hierarchy OQE*, *relation class OQE*. More expanded Jena Ontology-API oriented code versions are provided in subsection 3.2.4. The document text pattern matching *regular expression algorithm* is also shown and the modified *tf-idf* document relevance-scoring algorithm will be briefly referenced.

The objective of the query expansion algorithms is to identify only classes and individuals that have either an inheritance relationship with a query term, or have a wider relationship to the query term matching concept. To recap, *S+S+R OQE* implies *S+S OQE* plus any relation classes (i.e. defined by *asserted conditions*) for *every* class identified by *S+S OQE*. Only classes identified by *S+S OQE* are traversed to identify R classes.

Pseudo Code for Inheritance Class Hierarchy Algorithm

The inheritance class hierarchy algorithm generates query expansion terms depending on the selected class hierarchy expansion mode, i.e. *S OQE*, *S+S OQE* or *All OQE*. A super class *OQE* adds direct super class line classes, i.e. it excludes sub class branches. The algorithm will also identify equivalent classes and individuals – see Fig. 55.

```
for each ontology class $c$ {
  if $c$ subclass $c_{sub}$ or $c$ superclass $c^{sup}$ OQE required {
    for each keyword {
       if $c$ equals keyword {
         if $c^{sup}$ required {do $c^{sup}$Proc. }
         if $c_{sub}$ required {do $c_{sub}$Proc and do $c_{sub}$individualProc. }
         do cProc. }
       if cProc {
         add $c$ to OQE array.
         for $c$  list equivalent classes $c^{eq}$ {
           add $c^{eq}$ to OQE array. } }
       if do $c^{sup}$Proc AND $c$ has $c^{sup}$ {
          for $c$ list $c^{sup}$ {
             set Top class equal to next $c^{sup}$.
             do $c^{sup}$individualProc.
             add $c^{sup}$ to OQE array.
             for $c^{sup}$  list equivalent classes $eq^{sup}$ {
                add $eq^{sup}$ to OQE array. } }
          if do $c^{sup}$individualProc {
            for Top class list individuals {
              add individuals to OQE array. } }
       }
       if do $c_{sub}$Proc AND $c$ has $c_{sub}$ {
          for $c$ list $c_{sub}$ { add $c_{sub}$ to OQE array. } }
       if do $c_{sub}$individualProc AND ($c^{sup}$individualProc NOT executed) {
          for $c$ list individuals { add individual to OQE array. } }
  } }
  else if ($c_{sub}$ AND $c^{sup}$ OQE) NOT required {
    add $c$ to OQE array. } // Get All classes
}

if ($c_{sub}$ AND $c^{sup}$ OQE) NOT required {
  for each ontology class list individuals {
    add individual to OQE array. } } // Get All individuals
```

Fig. 55. High-level inheritance class hierarchy *OQE* algorithm.

The algorithm can be explained briefly: first, the algorithm checks to see if sub class and super class query expansion is required and then checks each base query term against the ontology classes; adding each direct match to the *OQE* array, together with equivalent classes and sub and super classes of the direct match concept. If the whole ontology is required, all classes are added to the *OQE* array. Both options add respective class individuals to the *OQE* array.

Expanded Jena API-based pseudo code is provided in subsection 3.2.4, Fig. 58.

Pseudo Code for Relation Class Algorithm

To recap, *S+S+R OQE* relation classes form the object o component in the binary relationship p(s, o) that is formed in an asserted condition, i.e. when an OWL object property p is specified to describe a class. The relation class algorithm, in Fig. 56, generates additional query expansion terms when *S+S+R OQE* is used. The algorithm only adds those relation classes belonging to any subject s class found by the *S+S OQE inheritance class hierarchy* algorithm.

```
for each ontology class $c$ {
  for each OQE array term where $c$ equals OQE array term {
    for each $c$ anonymous $c^{sup}$ list object property values $p_v$ {
      if $p_v$ NOT (null OR Resource OR Restriction OR Class) {
        add $p_v$ to PV array. } } } } // relation class?

for each PV array $p_v$ {
  for every $c$  where $p_v$ equals $c$ {
    if vector does not contain $c$ { // relation class
      add $c$ to vector and add $c$ and weight to OQE array. } } }
```

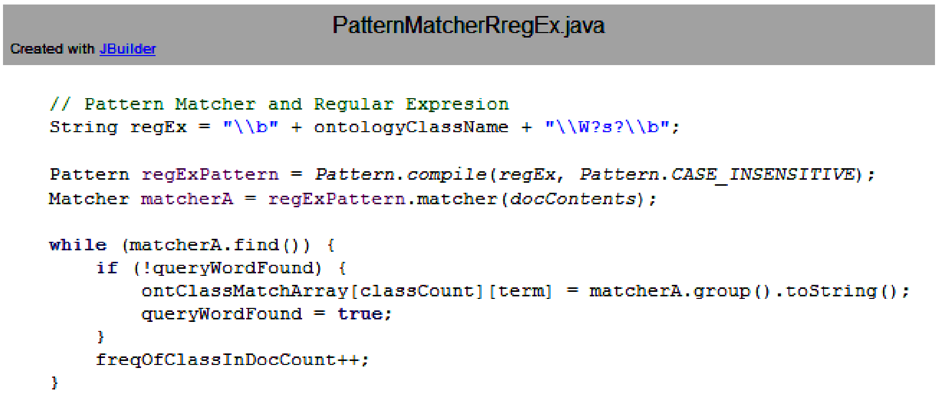
Fig. 56. High-level relation class *OQE* algorithm.

The algorithm works in two stages. The first stage matches each *OQE* array term with the full ontology class hierarchy and then uses the Ontology API methods to extract the property values of each anonymous class, for each of the *OQE* array terms; the property values represent the property and object class values of the asserted conditions specified. The second stage then compares the property values with the ontology class hierarchy, to identify incremental object (relation) classes and add them to the *OQE* array.

Expanded Jena API-based pseudo code is provided in subsection 3.2.4, Fig. 59.

Pattern Matching Regular Expression Algorithm

The pattern matching regular expression algorithm, for comparing an ontology query expansion term to document text, is shown in the Java code extract in Fig. 57.



PatternMatcherRregEx.java

Created with JBuilder

```java
// Pattern Matcher and Regular Expresion
String regEx = "\\b" + ontologyClassName + "\\W?s?\\b";

Pattern regExPattern = Pattern.compile(regEx, Pattern.CASE_INSENSITIVE);
Matcher matcherA = regExPattern.matcher(docContents);

while (matcherA.find()) {
    if (!queryWordFound) {
        ontClassMatchArray[classCount][term] = matcherA.group().toString();
        queryWordFound = true;
    }
    freqOfClassInDocCount++;
}
```

Fig. 57. Extract of Java pattern match and regular expression code.

The regular expression can handle word variations (e.g. ship, ships, ship’s, and –ship) to ensure that term counts are not overstated in the relevance algorithms.

The SemSeT search algorithm achieved accurate keyword/concept matching using the regular expression; this provides a basis for further tool functionality development.

Vector Space Model *tf-idf* Document Relevance Algorithm

The *tf-idf* document relevance algorithm was discussed in subsection 1.6.2; however, it requires modification to incorporate ontology concept relevance weightings, which will be discussed in subsection 3.2.5. As a preliminary explanation, the algorithm will simply multiply the frequency *F* of a term *i* in a document *d*, i.e. $F_{t_i\,d}$, by the concept weight O_w, i.e. $F_{t_i\,d} * O_w$ as shown next.

$$W_{td} = \sum_{t_i \,\in\, d,\; d \in D} \left(\frac{F_{t_i\,d} * O_w}{\max F_{td}} \right) * \ln \left(\frac{D}{n_{t_i}} \right)$$

The algorithm Java code, based on Jena Ontology API methods, will be briefly explained in Appendix D, by on matching key Java code variables with the *tf-idf* elements.

3.2.4 Extended Pseudo Code for Key *OQE* Algorithms

Expanded Jena Ontology API-related algorithms, for the key *OQE* pseudo code algorithms shown in subsection 3.2.3, are provided below; Appendix B shows Java syntax versions using the Jena API.

Extended Inheritance Class Hierarchy Algorithm Pseudo Code

The code shown below, in Fig. 58, represents the inheritance class hierarchy *OQE* algorithm oriented towards the Jena Ontology-API library; it extends the high-level pseudo code algorithm shown in subsection 3.2.3, Fig. 55.

Jena methods are used to break out operands for intersection and union classes. Also, it was necessary for the algorithm to be able to identify label names of classes (real world name for ontology class name where class names contained multiple joined words), to match real world query terms, e.g. to match a query term “Power Station”, class PowerStation would be specified with a label name “Power Station”.

```
for ontology iterator list named_classes C {
  if C sub_C OR C super_C required {//SUB/SUPER CLASS
    for each keyword {
      if C Label OR LocalName equals keyword {
        if super_C is required {
          do_super_C_proc; }
        if sub_C is required {
          do_sub_C_proc;
          do_sub_C_individuals_proc; }
        do_C_proc; }
     if do_C_proc {
       do_OQE_proc; // Do Named Class Proc
       for C equivalent_C iterator list equivalent_classes  EQ{
          if EQ is NOT intersection_C {
            do_OQE_proc; }
          else if EQ is intersection_C {
             for equivalent_C intersection_C iterator list intersection_C members {
                do_OQE_proc; } }
     } }
     if do_super_C_proc AND named_C has super_class {//Get Super Cs
       for C super_C iterator list super_classes SC {
         if SC is not anonymous_C {
           if SC is not a Restriction_C AND
                  not a Thing_C AND not a Resource_C {
              set Top_Class equal to next super_class;
              do_super_C_individuals_proc;
              do_OQE_proc;
              for super_C equivalent_C iterator list equivalent_classes  EQ{
                if EQ is NOT intersection_C {
                  do_OQE_proc; }
                else if EQ is intersection_C {
                  for equivalent_C intersection_C iterator list intersection_C members {
                do_OQE_proc; } } }
         } }
         else if SC is anonymous_C {
            if SC is union_C {
               for super_C union_C iterator list union_C members {
               do_OQE_proc; } }
            else if SC is intersection_C {
               for super_C intersection_C iterator list intersection_C members {
                  do_OQE_proc; } }
       } }
       if do_super_C_individuals_proc {
         for ontology iterator list named_classes CI {
            if CI Label or LocalName equals Top_Class {
              for CI individuals iterator list individuals {
              do_OQE_proc; } } } }
     }
     if do_sub_C_proc AND C has a sub_C {
        for C sub_C iterator list sub_classes {
          do_OQE_proc; } } // Get SUB CLASSES
     if do_sub_C_individuals_proc AND
                  (do_super_C_individuals_proc NOT executed) {
        for C individuals iterator list individuals {
           do_OQE_proc; } } // Get INDIVIDUALS
  } }
  else if sub_C NOT required AND super_C NOT required {
     do_OQE_proc; } // Get ALL Ontology CLASSES
}
if sub_C NOT required AND super_C NOT required {
  for ontology iterator list individuals {
    do_OQE_proc; } } // Get ALL Ontology INDIVIDUALS

do_OQE_proc { // Populate OQE array
  add class/individual Label to OQE array;
  add class/individual LocalName to OQE array; }
```

Fig. 58. Ontology class hierarchy and individuals indexing algorithm.

Extended Relation Class Algorithm Pseudo Code

The code shown below, in Fig. 59, identifies asserted condition relation classes and is based on the Jena API library; it extends the high-level pseudo code algorithm shown in subsection 3.2.3, Fig. 56. Also, the procedure has been improved to identify further relevant classes; during black-box testing it was established that the program failed to identify situations where a relation class also had an equivalent class. The algorithm therefore contains a second iteration of ontology class comparison to property value array class/terms, to then permit the ontology class iterator to list previously unidentied equivalent classes.

```
// add Relation classes belonging to the inheritance class hierarchy OQE classes

for ontology iterator list named_classes C { //Get ont classes
  add C LocalName and Label to ONT array;
  for each existing OQE array term {
    if C equals OQE array term { //matched term
      for C super_C iterator list super_classes {
        if super_C is anonymous {
          //Get asserted condition
          for super_C_property_value iterator list property_values $p_v$ {
            if  $p_v$ LocalName does NOT equal null OR "Resource" OR "Restriction" OR "Class" {
              // these are possible relation classes?
              add $p_v$ LocalName to property_value array;
} } } } } } }

for each property_value array item value $i_v$  {
  for each existing ONT array class/term ct {
    if $i_v$ equals ct {
      do_OQE_proc;
} } }

// Add any Relation class Equivalent classes
for ontology iterator list named_classes ct { //Get ont classes
  for each property_value array item value $i_v$  {
    if $i_v$ equals ct {
      for ct equivalent_C iterator list equivalent_classes  EQ{
         if EQ is NOT intersection_C {
            do_OQE_proc; }
         else if EQ is intersection_C {
             for equivalent_C intersection_C iterator list intersection_C members {
                do_OQE_proc; } }
} } } }

do_OQE_proc {   // Populate OQE array
   // add only new class/terms
   if vector does not contain ct {
      add ct to vector;
      // add new relation class to OQE term set
      add ct LocalName and Label to OQE array;
} }
```

Fig. 59. Identification of relation classes created by asserted conditions.

It should be noted that some problems were experienced when attempting to develop the algorithm to read union-based asserted conditions, i.e. using relations A hasRelation (X ⊔ Y); it became necessary to modify the OWL syntax, to keep X and Y assertions separate, as it was not possible to list union operands using Jena Ontology API methods.

3.2.5 Formulation of Concept (Term) Relevance Weights

It has been highlighted that the semantic correlation between ontology concepts can be reflected as a degree of relevance that can be applied to a query expansion concept based on its semantic distance from a query term (Fang et al., 2005, Gligorov et al., 2007, Tiun et al., 2001, Rocha et al., 2004, Bhogal et al., 2007). To recognise semantic distance and relationship between ontology concepts, the ranking algorithm was refined to incorporate query expansion relevant class weightings – similar to (Fang et al., 2005), i.e. keyword-related ontology class weightings O_w were applied to allow SemSeT's *OQE* results to reflect a concept's position in the semantic hierarchy. Fig. 60 illustrates how this relative weighting system was applied, based on any class matching a keyword being first awarded a weighting of 1.0, e.g. as class type Ship is a super class of the keyword matching class CargoShip, Ship was ranked lower (0.7) than CargoShip (1.0). In turn, sub class Tanker (0.3) was weighted lower CargoShip. The initial rationale for such weights was that a CargoShip is always a Ship but not necessarily a Tanker, therefore a super class could be weighted above a sub class.

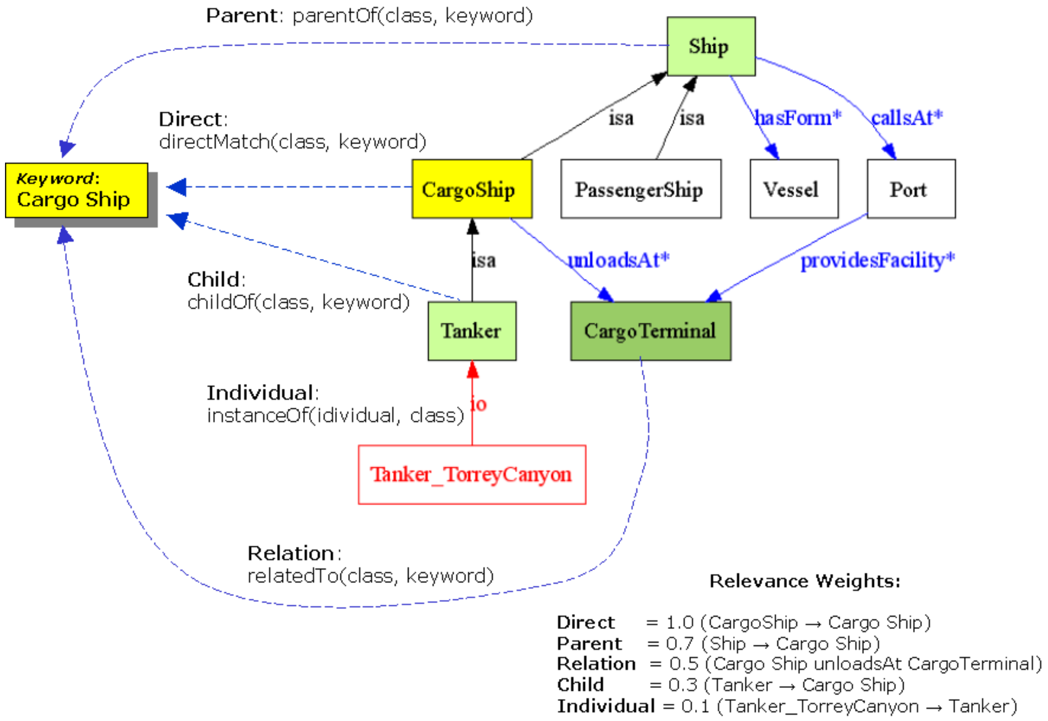

Parent: parentOf(class, keyword)
Ship
Direct:
directMatch(class, keyword)
isa
isa
hasForm*
callsAt*
Keyword:
Cargo Ship
CargoShip
PassengerShip
Vessel
Port
isa
unloadsAt*
providesFacility*
Child:
childOf(class, keyword)
Tanker
CargoTerminal
Individual:
instanceOf(idividual, class)
io
Tanker_TorreyCanyon
Relation:
relatedTo(class, keyword)
Relevance Weights:
Direct = 1.0 (CargoShip → Cargo Ship)
Parent = 0.7 (Ship → Cargo Ship)
Relation = 0.5 (Cargo Ship unloadsAt CargoTerminal)
Child = 0.3 (Tanker → Cargo Ship)
Individual = 0.1 (Tanker_TorreyCanyon → Tanker)

Fig. 60. Semantic distance relevance weights.

Other weightings can be applied, e.g. individual (Tanker_TorreyCanyon) of class type Tanker might be given a nominal weight (0.1). However, research has also identified that improved retrieval results can be achieved by recognising the contextual relevance of class types having a wider semantic domain relationship, as opposed to having a direct inheritance line with the keyword matching class, e.g. queries expanded with WordNet® synset term *glosses* (Navigli and Velardi, 2003). Gloss terms were considered more useful than relying on hypernyms and hyponyms, similar to the term clustering and co-occurrence characteristics of probabilistic theory. Such *relation classes*, described and identified in OWL by asserted conditions, e.g. CargoShip unloadsAt CargoTerminal, could be weighted 0.5. Interestingly, DL union structured asserted conditions, i.e. Z hasRelation (X ⊔ Y), were problematic in the *S+S+R OQE* algorithm and had to be defined individually – see subsection 3.2.6 (Context for 'T416 Hydro-electric').

Using the suggested weighting approach, an enhanced *document* weight vector W^{+}_{d} can now be created by modifying the *term* weight vector presented earlier in subsection 1.6.2, i.e. by multiplying the frequency *F* of any matched term t_i in document *d*, with the term's related concept weighting, i.e. $F_{t_i d} * O_w$. The result is:

$$W_{td} = \sum_{t_i \in d,\, d \in D} \left(\frac{F_{t_i d} * O_w}{\max F_{td}} \right) * \ln \left(\frac{D}{n_{t_i}} \right)$$

The objective of this algorithm modification is therefore to recognise the degree of relevance of classes that either relate to *S+S OQE* mode or *S+S+R OQE* mode classes. However, as the weight allocations were accepted as being fairly subjective, further examination was required and the term weightings were subsequently modified on some later experiments – see further experimentation results - subsections 4.4.1 to 4.4.3.

3.2.6 Design of Ontology Search Contexts

The TREC WT2g corpus contains 50 topics, from which, three topics (401 "Foreign Minorities, Germany", 416 "Three Gorges Project" and 438 "Tourism, increase") were selected for the query experiments, with each providing the basis for query matrix and ontology context formulation. For convenience, they will be referred to as T401, T416 and T438.

A number of contexts were created for the *OQE* experiments; some were used as individual contexts and some as imports for a wider, multi-context subject domain. T438 used a Tourism ontology context, widened by importing various ontology contexts, and embracing over 650 concepts. T401 used an Immigration ontology context, covering 41 concepts and T416 the Hydro-electric context, covering 58 concepts; these are discussed in the following subsections and form the basis for the *OQE* experimentation results comparisons in chapter 4. The different ontology designs will vary between shallow hierarchies and a deeper hierarchy. Complete examples of the various ontology class hierarchies are provided in Appendix C.

Ontology Context for T401 'Foreign minorities, Germany'

Fig. 61 shows the T401 topic statement query guidelines used to create an Immigration context ontology. The query description is "What language and cultural differences impede the integration of foreign minorities in Germany?" A relevant document is defined as one being related to Germany and focusing on the causes of the lack of integration; not just immigration problems. Google and Swoogle were used to identify Immigration ontology sources and the primary sources were The Home Office Border Agency site [1] and glossary of terms [2].

<num> Number: 401

<title> foreign minorities, Germany

<desc> Description:

What language and cultural differences impede the integration of foreign minorities in Germany?

<narr> Narrative:

A relevant document will focus on the causes of the lack of integration in a significant way; that is, the mere mention of immigration difficulties is not relevant. Documents that discuss immigration problems unrelated to Germany are also not relevant.

Fig. 61. Topic statement for T401 query experiment

The Immigration context has a shallow hierarchy, which limits hierarchical query expansion using *S+S OQE*. An extract of the ontology, created with the Protégé OWLViz graphic tool, is shown in subsection 3.3.1, Fig. 64.

Ontology Context for T416 'Three Gorges Project'

The Three Gorges Project is a major hydro-electric scheme in China; consequently, Fig. 62 shows the T416 topic statement query guidelines used for a Hydro-electric context ontology.

<num> Number: 416

<title> Three Gorges Project

<desc> Description:

What is the status of The Three Gorges Project?

<narr> Narrative:

A relevant document will provide the projected date of completion of the project, its estimated total cost, or the estimated electrical output of the finished project. Discussions of the social, political, or ecological impact of the project are not relevant.

Fig. 62. Topic statement for T416 query experiment.

The description of the T416 query is "What is the status of The Three Gorges Project?" The guideline is that a relevant document will show the date of project completion, estimated total cost, or the estimated electrical output. Social, political, or ecological issues are not relevant. A Hydro-electric ontology was developed for query expansion.

[1] Border Agency: http://www.bia.homeoffice.gov.uk/.

[2] Border Agency Glossary: http://www.ukba.homeoffice.gov.uk/glossary.

Google was used to identify contextually related primary reference points for concepts - the British Dam Society site [1] and Wikipedia's Three Gorges Dam content [2].

An extract of the class hierarchy is shown in subsection 3.3.2, Fig. 65. The ontology is similar to the Immigration ontology, i.e. it demonstrates a shallow hierarchy with a limited potential for *S+S OQE* ; therefore more detailed class descriptions expressivity was embedded in the context, to exploit *S+S+R OQE*.

As mentioned in subsection 3.2.4, "Extended Relation Class Algorithm", Protégé was used to initially create OWL syntax union structured asserted conditions, i.e. Z hasRelation (X ⊔ Y), unfortunately, attempts to extract union operands in the relation class *OQE* algorithm, using Jena Ontology API methods, were found to be problematic and were therefore avoided; consequently, X and Y assertions had to be separated.

Ontology Context for T438 'Tourism, increase'

Fig. 63 shows the T438 topic statement used for queries made via an existing Tourism context ontology, which also contained a number of imported tourism linked contexts.

<num> Number: 438

<title> tourism, increase

<desc> Description:

What countries are experiencing an increase in tourism?

<narr> Narrative:

A relevant document will name a country that has experienced an increase in tourism. The increase must represent the nation as a whole and tourism in general, not be restricted to only certain regions of the country or to some specific type of tourism (e.g., adventure travel). Documents discussing only projected increases are not relevant.

Fig. 63. Topic statement for T438 query experiment.

T438's query description is "What countries are experiencing an increase in tourism?" The somewhat *general* query guideline indicates that relevant documents will name a country having experienced increased tourism as a whole, i.e. nationwide and not regionally, based on tourism in general as opposed to a specific type of tourism, with documents discussing only increase projections being not relevant. Given the statement, it was considered that this topic might offer less opportunity for constructing precise queries.

[1] British Dam Society: http://www.britishdams.org/.

[2] Wikipedia: http://en.wikipedia.org/wiki/Three_Gorges_Dam.

This ontology is markedly larger (653 concepts, of which 81 are directly Tourism) than the T401 (41) and T416 (58) ontologies. T438 is larger because it contains a number of ontology imports that were considered context related to Tourism (including Commerce, Culture, Entertainment, Food, Road, Rail, Air, Sea, PopGroup, Retail, Sport) and was created during earlier research activity, i.e. before the TREC corpus had been identified; thus it presented an opportunity to *reuse* an existing ontology - even though it had not been developed in the context of the T438 topic. Some of the primary sources for the specific Tourism classes were the International Ecotourism Society (IES) [1], the Tourism Network [2] and its associated introductory guide (TN Guide) [3]. An extract of the main class hierarchy of directly Tourism-related concepts is shown in subsection 3.3.3, Fig. 67.

Given the multi-contextual nature of the Tourism ontology and number of classes, a comparison with the Immigration and Hydro-electric ontology contexts suggested that the slightly deeper class hierarchy might provide an opportunity to conduct more meaningful *S+S OQE*, although the less-specific nature of the topic statement might compromise *OQE* search effectiveness.

3.3 HOW THE EXPERIMENT WAS IMPLEMENTED

This section discusses the query approach, based on the TREC corpus and the search tool's *OQE*-enabling ontologies. The retrieval experiments make use of TREC WT2g independent data, i.e. supported by a set of query relevance judgements for each query topic. The query topic statements, and the ontology contexts, provide the basis for query matrix formulation.

A review of the corpus document list revealed that the document search pools for the three selected topics, T401 (Foreign Minorities, Germany), T416 (Three Gorges Project) and T438 (Tourism, increase), were spread unevenly throughout the corpus. Further, as SemSeT did not support document indexing and the documents had to be read at each query, it was decided to truncate searches, at optimal points, to avoid extended processing time, i.e. adopting a "Pareto" approach to obtain an optimal representative search pool without excessive document search. This allowed for 37 of the 45 relevant documents to be covered in a pool of 13,065 documents for T401; for T416, the statistics were 10 out of 14 relevant documents in a pool of 160,838 documents and, for T438, 36 out of 46 relevant documents in a pool of 96,885.

The decision to search a representative number of document folders was validated by conducting comparison searches for 3 selected queries for each of the 3 TREC topics, using all 247,491 documents in the corpus.

[1] IES: http://www.ecotourism.org/.

[2] Tourism Network: http://www.tourismnetwork.co.uk/.

[3] TN Guide: http://www.tourismnetwork.org/Tourism_Network_Intro_Guide_to_Tourism.pdf.

The resulting 9 queries were each executed on the basis of comparing keyword to *S+S* and *S+S+R OQE*s, using *optional* query terms (Ko, Oo and Oro options). The purpose of the comparison was to see if the *relative* performance of the Ko, Oo and Oro P&R profiles changed markedly, when the truncated document sets were searched, compared to searches on the full document collection. The results based on an average of the P&R profile percentages, across the three topics, are shown in Fig. 68 and Fig. 69 at the beginning of chapter 4; they show similar P&R profiles between the full and truncated document sets.

For the three main TREC topic experiments, it is perhaps worth restating: T438 used a Tourism ontology context, expanded with various Tourism-linked contexts, and embracing over 650 concepts (81 directly Tourism). T401 used a smaller Immigration ontology context, covering 41 concepts, and T416 employed the Hydro-electric context of 58 concepts. The variation in ontology components will range between shallow hierarchy *OQE*, using a mix of *S+S OQE*, *All OQE* and *S+S+R OQE*, compared to deeper hierarchy *OQE*, using solely *S+S OQE*. The ontology class hierarchies are provided in full in Appendix C.

Each query, in the 3 main experiments, involves a query set of query term search options (i.e. Ko, Oo, Km and Om in T401 and T438, with Ko, Oo, Orm, Km, Om and Orm in T416).

As mentioned previously and demonstrated in subsection 3.1.4, each query is derived from the four base query terms used in keyword search mode (Ko or Km options). The matrix query term selection approach was to emulate how keywords/query terms might possibly be applied when using a *non-OQE* search interface. The following query matrices were not devised to favour *OQE* by ensuring a *convenient* spread of ontology context terms that would secure the greatest or optimal query expansion; some query term combinations result in some of the individual terms generating duplicate expansions, e.g. T416 Q5, T438 Q12 and Q13.

3.3.1 T401 ‘Foreign minorities, Germany’

T401 *OQE* experiments were completed using the following Immigration ontology context and query matrix.

T401 Immigration Ontology Context

An extract of the Immigration context’s class hierarchy is shown in Fig. 64 and it shows that it has a fairly shallow class hierarchy, i.e. viewed left to right; the ontology has 41 concepts in total and therefore only limited traversal using *S+S OQE* was found to be achievable.

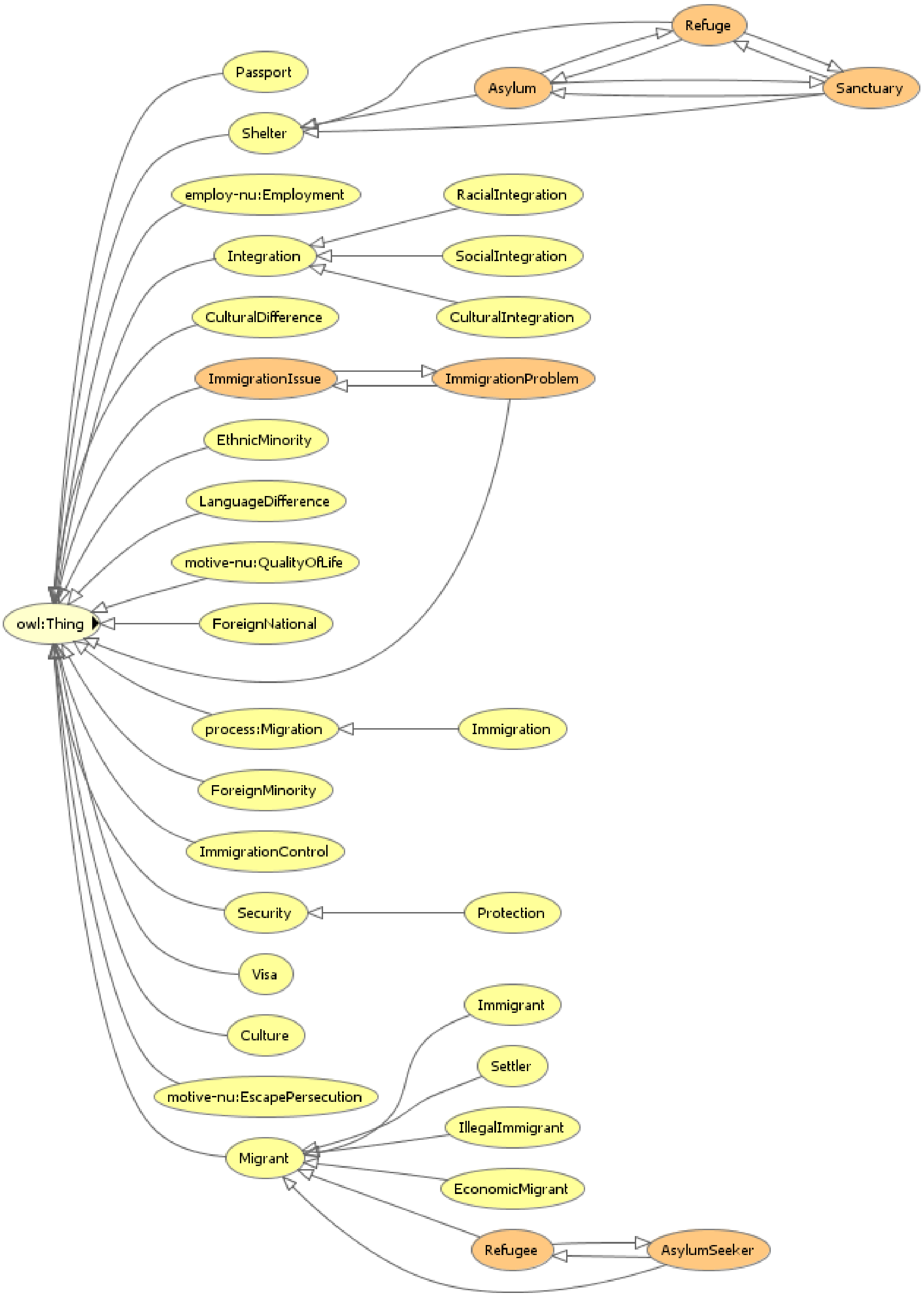
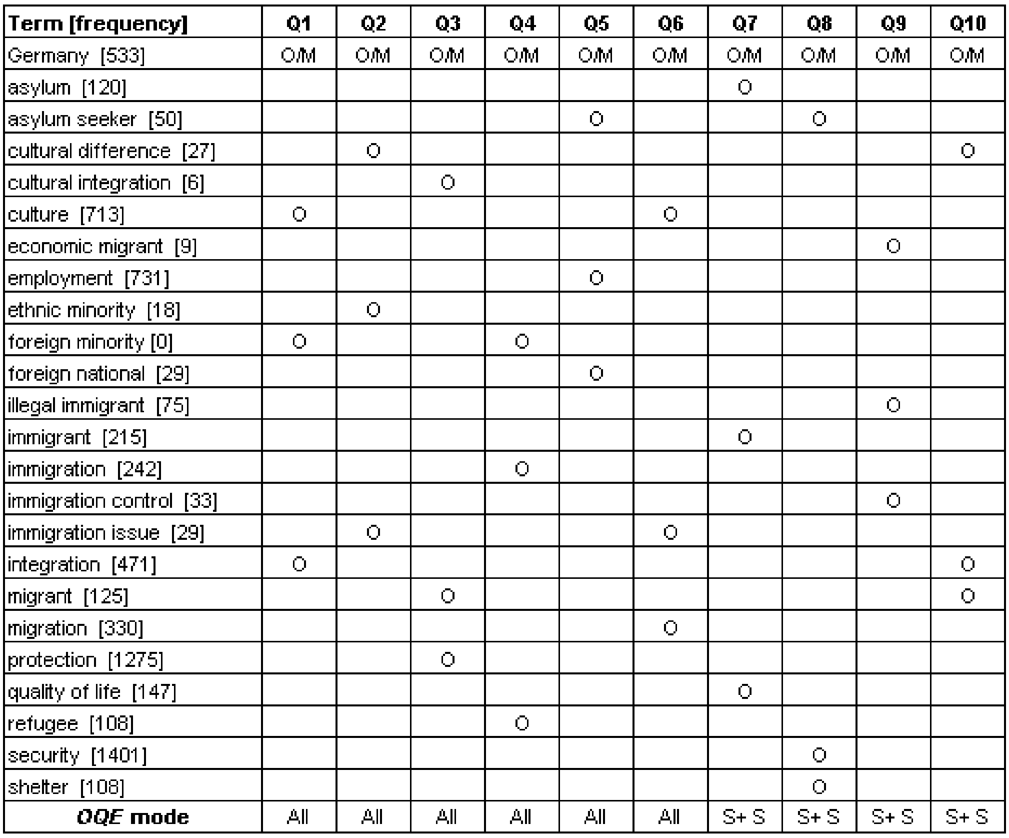


Fig. 64. Extract of the Immigration context.

T401 Query Matrix

A query matrix was created based on the T401 topic statement in subsection 3.2.6 and the available Immigration context concepts. The matrix embraced 24 terms across 10 query comparison sets. For T401, the key driver for query matrix was the query topic's description "What language and cultural differences impede the integration of foreign minorities in Germany?" Given that the term "Germany" had to feature in relevant documents, it was adopted as a constant in each of the 10 query sets, i.e. as an *optional* term in Ko, Oo query term searches and as the *must-have* term in the Km and Om searches. Table 8 shows the matrix query term combinations, which were an attempt to reflect the TREC query guidelines, i.e. by focusing on the causes of the lack of integration. O/M denotes the term has been used as both an *optional* and *must-have* term; O denotes an *optional* query term.

Table 8. TREC 401 Foreign Minorities query matrix.



Term [frequency]	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**
Germany [533]	O/M	O/M	O/M	O/M	O/M	O/M	O/M	O/M	O/M	O/M
asylum [120]							O			
asylum seeker [50]					O			O		
cultural difference [27]		O								O
cultural integration [6]			O							
culture [713]	O					O				
economic migrant [9]									O	
employment [731]					O					
ethnic minority [18]		O								
foreign minority [0]	O			O						
foreign national [29]					O					
illegal immigrant [75]									O	
immigrant [215]							O			
immigration [242]				O						
immigration control [33]									O	
immigration issue [29]		O				O				
integration [471]	O									O
migrant [125]			O							O
migration [330]						O				
protection [1275]			O							
quality of life [147]							O			
refugee [108]				O						
security [1401]								O		
shelter [108]								O		
***OQE* mode**	All	All	All	All	All	All	S+ S	S+ S	S+ S	S+ S

Given the limited opportunity to conduct extensive *OQE* based on *S+S* manipulation, it was decided that the query group Q1 to Q6 would be based on comparisons of keyword queries against *OQE* using all classes in the ontology hierarchy, i.e. *All OQE* mode; this also provided a generalised query expansion approach as opposed to other *OQE* modes. To provide a result comparison mix, queries Q7 to Q10 would compare keywords against *OQE* limited to sub and super class hierarchy, i.e. *S+S OQE*. Table 9 shows examples of *OQE* terms for base keyword terms used in queries Q4, Q8 and Q10.

The resulting keyword to *OQE* term ratio is also shown for each query, e.g. Q4 resulted in a ratio of 4:41, i.e. the 4 base query terms generated 41 terms in *All OQE* mode.

Table 9. Expansions for queries Q4, Q8 and Q10.

Query 4		
Term	*All OQE*	
Foreign minority	Asylum Asylum Seeker Cultural Difference Cultural Integration Culture Economic Migrant Employment Escape Natural Disaster Escape Persecution Ethnic Minority Foreign Minority Foreign National Illegal Immigrant Immigrant Immigration Immigration Control Immigration Destination Immigration Issue Immigration Problem Immigration Quota	Integration Language Difference Migrant Migration Nationality Passport Protection Quality of Life Racial Integration Refuge Refugee Rejoin Family Member Right of Abode Sanctuary Security Settler Shelter Social Integration Stateless Visa
Immigration		
Refugee		
Germany	Germany	
Keyword to OQE ratio 4:41		

Query 8	
Term	*S+S OQE*
Asylum seeker	Asylum Seeker Refugee Migrant
Security	Security
Shelter	Shelter Protection Refuge Sanctuary
Germany	Germany
Keyword to OQE ratio 4:9	

Query 10	
Term	*S+S OQE*
Cultural difference	Cultural Difference
Integration	Integration Social Integration Racial Integration Cultural Integration
Migrant	Migrant Asylum Seeker Refugee Immigrant Illegal Immigrant Settler Economic Migrant
Germany	Germany
Keyword to OQE ratio 4:13	

3.3.2 T416 ‘Three Gorges Project’

T416 *OQE* experiments were executed using the following Hydro-electric ontology context, specified to provide relation class expansions for additional *S+S+R OQE*, and the query matrix.

T416 Hydro-electric Ontology Context

An extract of Hydro-electric’s class hierarchy is shown in Fig. 65. The ontology is similar to the Immigration ontology, i.e. it demonstrates a shallow hierarchy with only a limited *OQE* by *S+S* manipulation being possible, although Hydro-electric is around 45% larger, with 58 classes in total.

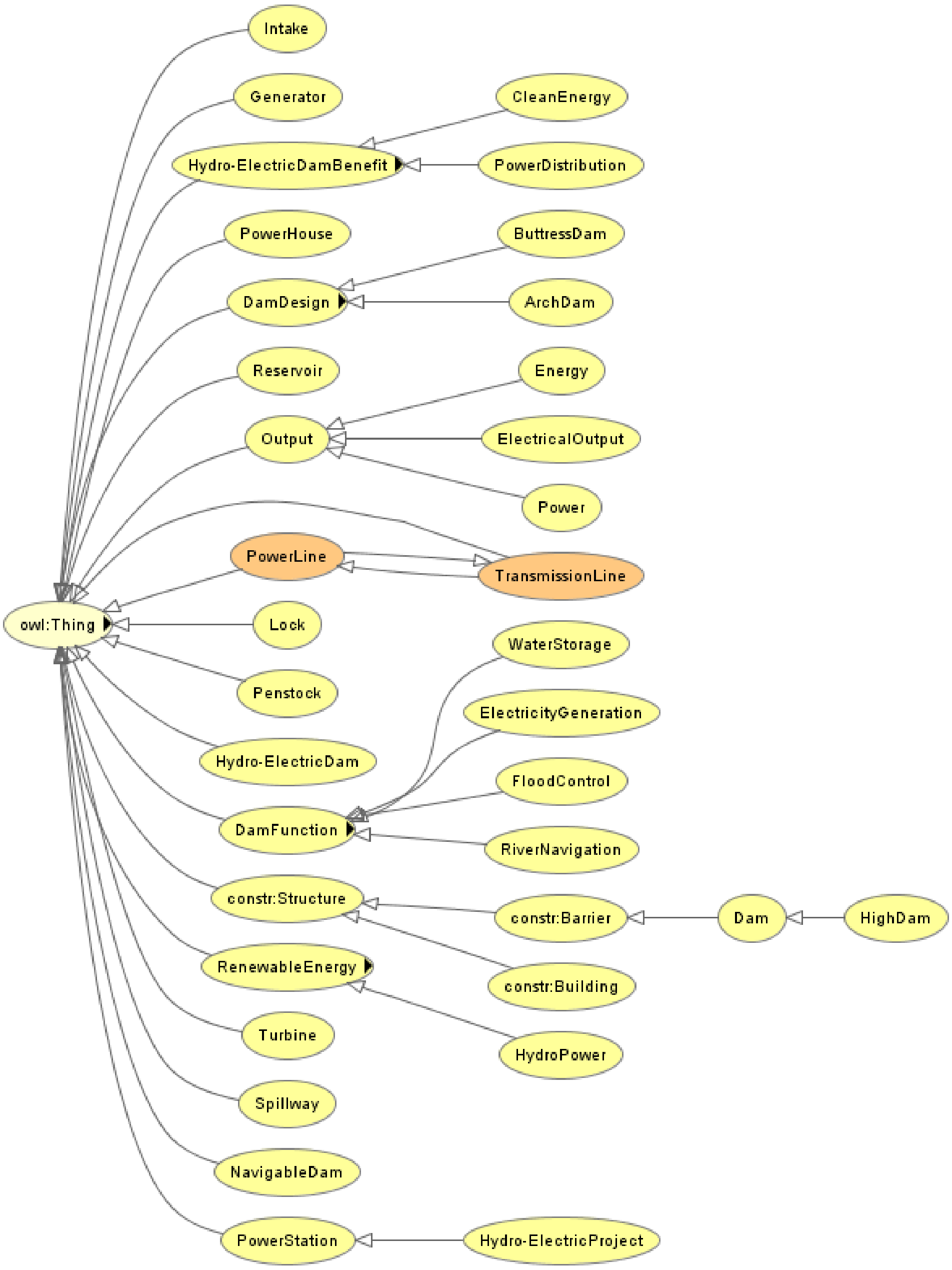


Fig. 65. Extract of the Hydro-electric context.

Fig. 66 shows the asserted conditions specified for use in additional query expansions made in *S+S+R OQE* mode.

For presentation purposes, asserted condition classes, i.e. those related to an object property to describe and constrain a class, are shown grouped together in the ellipses.

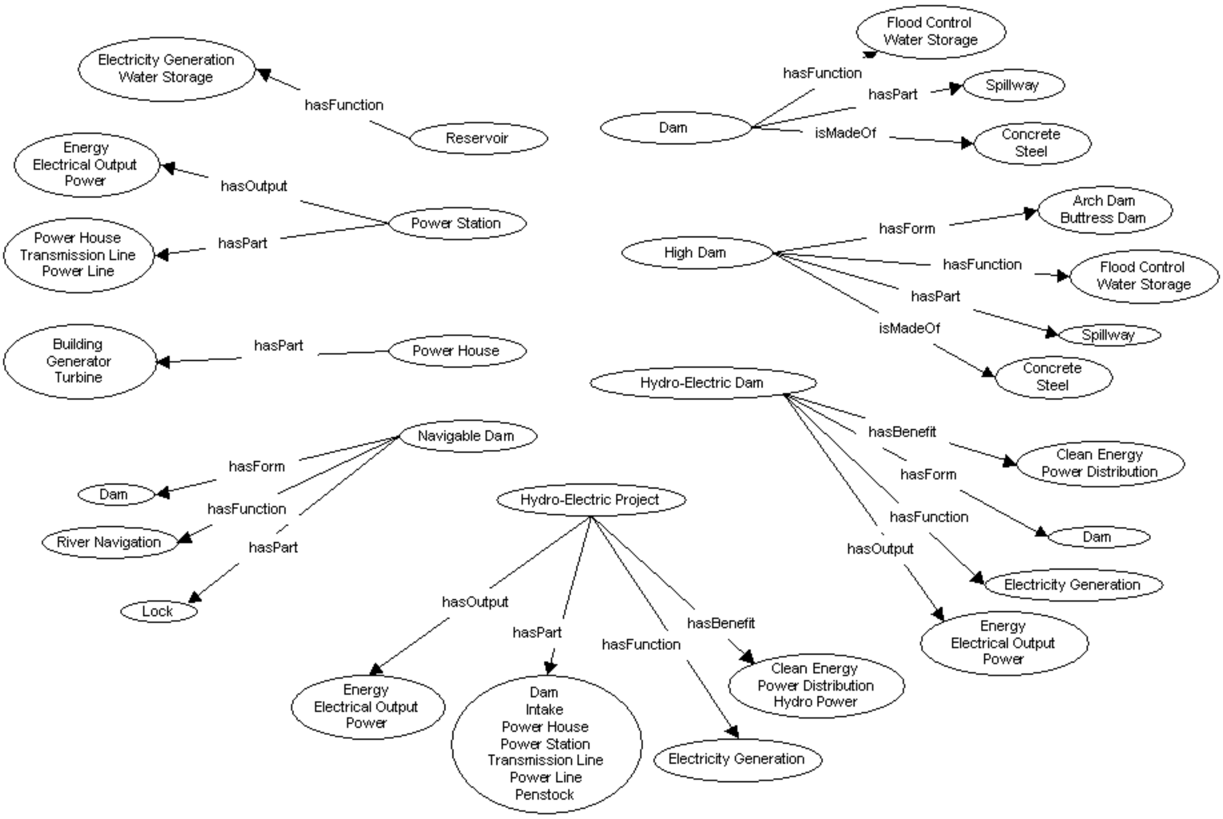


Fig. 66. Relations specified in the Hydro-electric context.

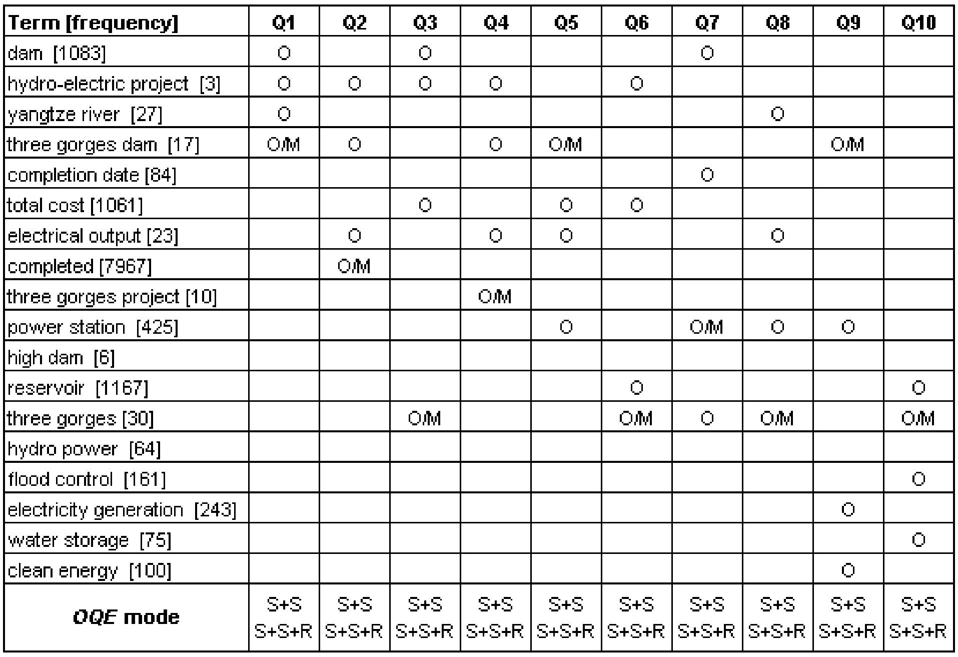
T416 Query Matrix

The T416 topic statement (subsection 3.2.6) and the developed ontology context, form the basis for the T416 query matrix in Table 10. The matrix used 18 base query terms over the 10 query sets. O/M denotes the query term used as both *optional* and *must-have*; O denotes an *optional* term.

As the term "Three Gorges Project" was the primary focus for relevant documents, this phrase and derivatives like "three gorges dam" were regularly used as anchor terms in the 10 query comparison groups, and for the *must-have* term. The matrix query term combinations sought to reflect the TREC query guidelines, i.e. to focus on the projected date of completion of the project, the estimated total cost or the estimated electrical output.

Again, given the hierarchical limitations of conducting extensive sub and super class *OQE*, it was decided adopt a three-way query comparison approach, i.e. based on comparisons of keywords against *S+S OQE* and then by comparing the two against *S+S+R OQE* (sub and super class expansion plus their specified relation classes). T416 therefore provides an alternative comparison approach to T401, in a sense being positioned between *S+S OQE* and *All OQE*.

Table 10. TREC 416 Three Gorges Project query matrix.



Term [frequency]	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10
dam [1083]	O		O				O			
hydro-electric project [3]	O	O	O	O		O				
yangtze river [27]	O							O		
three gorges dam [17]	O/M	O		O	O/M				O/M	
completion date [84]							O			
total cost [1061]			O		O	O				
electrical output [23]		O		O	O			O		
completed [7967]		O/M								
three gorges project [10]				O/M						
power station [425]					O		O/M	O	O	
high dam [6]										
reservoir [1167]						O				O
three gorges [30]			O/M			O/M	O	O/M		O/M
hydro power [64]										
flood control [161]										O
electricity generation [243]									O	
water storage [75]										O
clean energy [100]									O	
***OQE* mode**	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R	S+S S+S+R

Table 11 shows example *OQE*s for Q1, Q5, Q7 and Q10. The keyword to *OQE* is also shown, e.g. Q1 *OQE* ratio was 4:9:28 (4 keywords, 9 terms via *S+S OQE* and 28 via *S+S+R OQE*.

Table 11. Expansions for queries Q1 and Q5.

* *Relation classes*, i.e. related by asserted condition restriction				
Query 1			Query 5	
Term	*S+S, S+S+R* OQE*		Term	*S+S, S+S+R* OQE*
Dam	Dam Barrier Structure Three Gorges Dam High Dam *Spillway** *Steel** *Concrete** *Water Storage** *Flood Control** *Buttress Dam** *Arch Dam**		Three Gorges Dam	three gorges dam
Hydro-electric Project	Hydro-Electric Project Power Station Three Gorges Project *Power Line** *Power** *Transmission line** *Power House** *Electrical Output** *Energy** *Power Distribution** *Hydro Power** *Electricity Generation** *Dam** *Intake** *Clean Energy** *Penstock**		Power Station	Power Station Hydro-Electric Project Three Gorges Project *Power Line** *Power** *Transmission line** *Power House** *Electrical Output** *Energy** *Power Distribution** *Hydro Power** *Electricity Generation** *Dam** *Intake** *Clean Energy** *Penstock**
Yangtze River	Yangtze river		Electrical Output	Output Electrical Output
Three Gorges Dam	three gorges dam		Total Cost	total cost
Keyword to OQE ratio 4:9:28			*Keyword to OQE* ratio 4:7:19	

Table 11 (continued). Expansions for queries Q7 and Q10.

* _Relation classes_, i.e. related by asserted condition restriction			
Query 7		Query 10 (MORE GENERIC)	
Term	_S+S, S+S+R* OQE_	Term	_S+S, S+S+R* OQE_
Dam	Dam Barrier Structure Three Gorges Dam High Dam _Spillway*_ _Steel*_ _Concrete*_ _Water Storage*_ _Flood Control*_ _Buttress Dam*_ _Arch Dam*_	Reservoir	Reservoir Three Gorges Reservoir _Electricity Generation*_
Power Station	Power Station Hydro-Electric Project Three Gorges Project _Power Line*_ _Power*_ _Transmission line*_ _Power House*_ _Electrical Output*_ _Energy*_ _Power Distribution*_ _Hydro Power*_ _Electricity Generation*_ _Dam*_ _Intake*_ _Clean Energy*_ _Penstock*_	Three Gorges	three gorges
Three Gorges	three gorges	Flood Control	Flood Control Dam function
Completion Date	completion date	Water Storage	Water Storage Dam function
Keyword to OQE ratio 4:10:29		_Keyword to OQE_ ratio 4:6:7	

3.3.3 T438 ‘Tourism, increase’

T438’s Tourism ontology context contains imports of various Tourism-linked ontologies. The main Tourism classes and query matrix are shown below.

T438 Tourism Multi-Context Ontology

Examples of the directly Tourism-related classes, extracted from the multi-context Tourism ontology, are shown in the class hierarchy in Fig. 67.

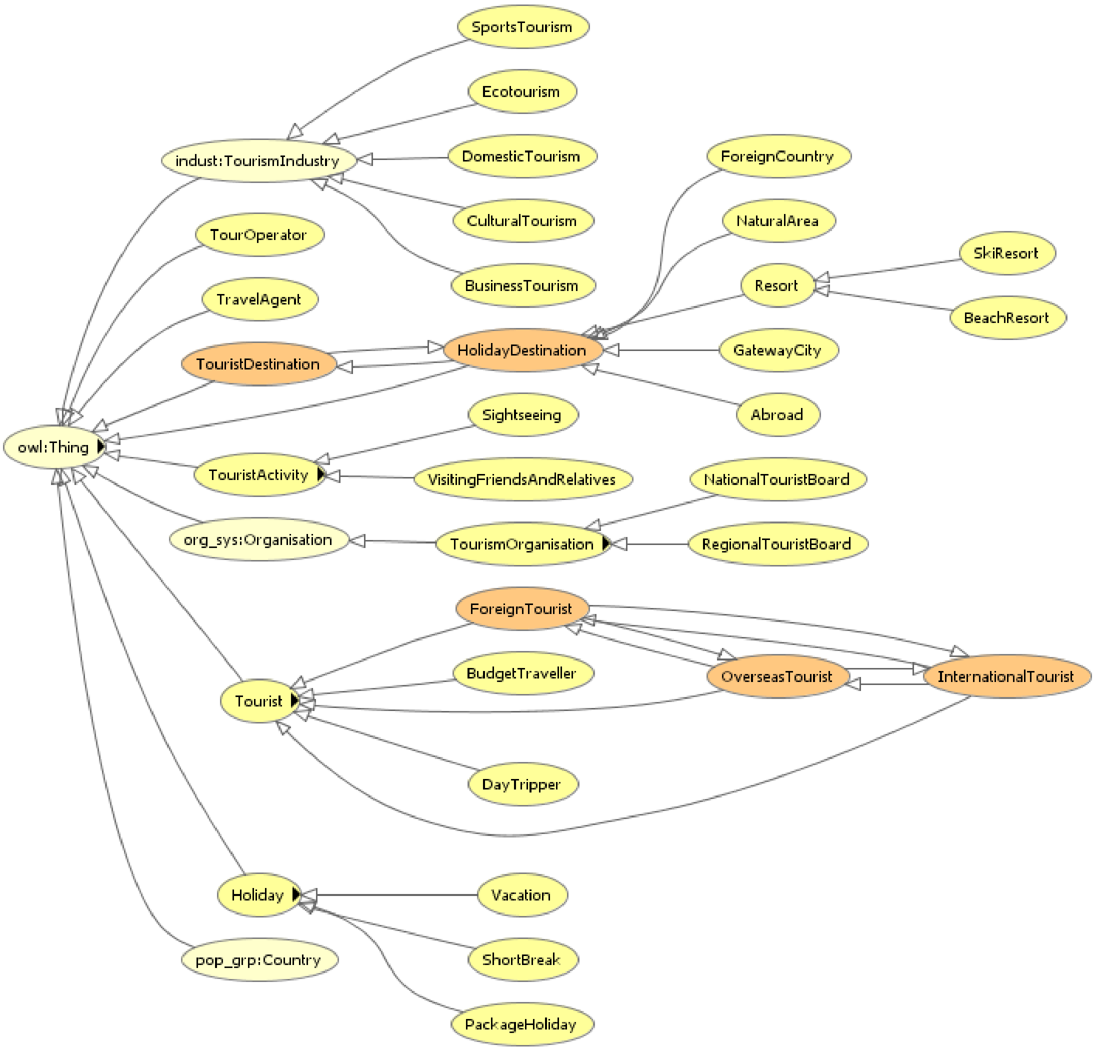


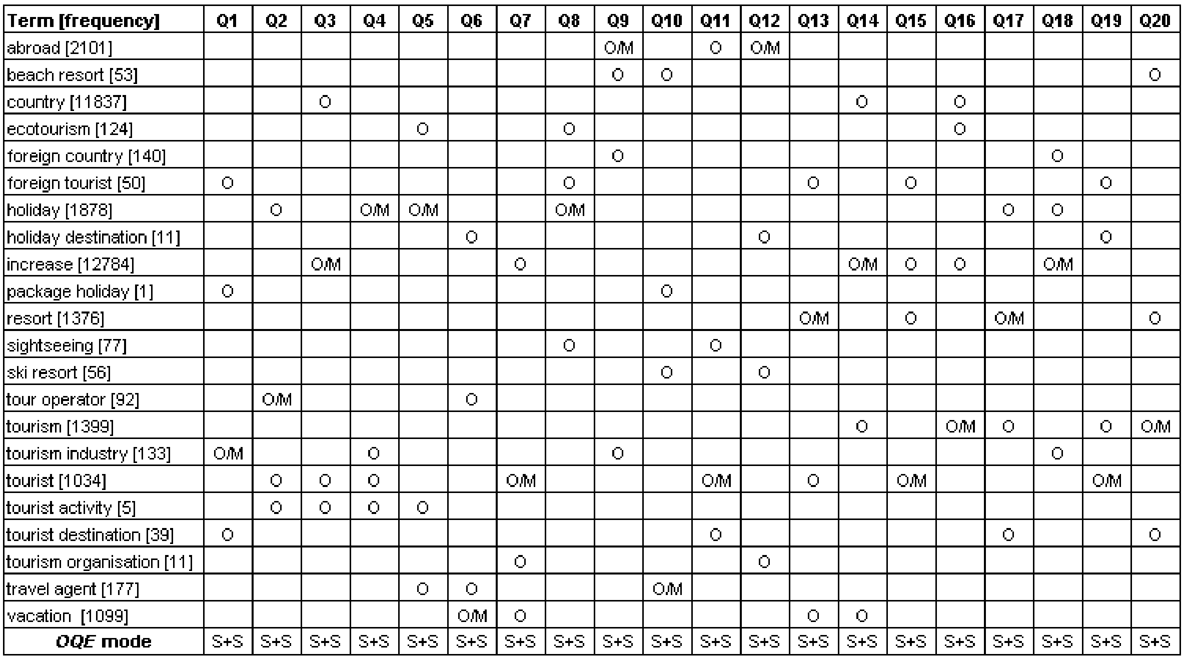
Fig. 67. Extract of the Tourism ontology.

In comparison to the Immigration and Hydro-electric contexts, the Tourism ontology presents a relatively deeper class hierarchy, although the non-specific query topic statement presented possible query limitations; nevertheless, it provides the potential for more extended *S+S OQE*.

T438 Query Matrix

The query matrix was developed using the T438 topic statement (subsection 3.2.6) query description "What countries are experiencing an increase in tourism?" and the *reused* Tourism ontology. The matrix query term combinations were an attempt to reflect the query objective of finding documents naming a country that has experienced a countrywide increase in tourism in general. However, it became evident that precise T438 queries could be more difficult to achieve than in the more specific T401 and 416 objectives, e.g. "Tourism and its increase" and the encapsulation of "increase" to "country as a whole" presented a greater query formulation challenge. It was also anticipated that IR results might be more problematic, given the reuse of an existing multi-context ontology. Consequently, a wider mix of query term combinations was applied, across 20 query comparison groups; this was particularly evident with the *must-have* anchor terms. The query matrix is shown in Table 12. O/M denotes both an *optional* and *must-have* query term; O denotes an *optional* term.

Table 12. TREC 438 Tourism query matrix.



Term [frequency]	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
abroad [2101]									O/M		O	O/M								
beach resort [53]									O	O										O
country [11837]			O											O		O				
ecotourism [124]					O			O								O				
foreign country [140]									O									O		
foreign tourist [50]	O							O					O		O				O	
holiday [1878]		O		O/M	O/M			O/M									O	O		
holiday destination [11]						O						O							O	
increase [12784]			O/M				O							O/M	O	O		O/M		
package holiday [1]	O									O										
resort [1376]													O/M		O		O/M			O
sightseeing [77]								O			O									
ski resort [56]										O		O								
tour operator [92]		O/M				O														
tourism [1399]														O		O/M	O		O	O/M
tourism industry [133]	O/M			O					O									O		
tourist [1034]		O	O	O			O/M				O/M		O		O/M				O/M	
tourist activity [5]		O	O	O	O															
tourist destination [39]	O										O						O			O
tourism organisation [11]							O					O								
travel agent [177]					O	O				O/M										
vacation [1099]						O/M	O						O	O						
***OQE* mode**	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S	S+S

Tourism is a deeper and wider ontology, with the opportunity for the group of 20 query sets to be based on keywords versus *S+S OQE*, therefore only this *OQE* mode was used. Table 13 shows expansions for queries Q1, Q4, Q8 and Q12.

Table 13. Expansions for queries Q1 and Q4.

Query 1			Query 4		
Term	*S+S OQE*		Term	*S+S OQE*	
Foreign tourist	Foreign Tourist Overseas Tourist International Tourist Tourist		Holiday	Holiday Activity Holiday Package Holiday Sun Sea and Sand Holiday Cruise Holiday	Vacation Short Break Fly-Cruise Holiday City Break
Package holiday	Package holiday Holiday		Tourism industry	Tourism Industry Cultural Tourism Ecotourism	Domestic Tourism Sports Tourism Business Tourism
Tourism industry	Tourism Industry Cultural Tourism Ecotourism Business Tourism Domestic Tourism Sports Tourism		Tourist	Tourist Budget Traveller Day Tripper Business Tourist Overseas Tourist	Back Packer International Tourist Domestic Tourist Day Visitor Leisure Tourist Foreign Tourist
Tourist destination	Tourist destination Holiday destination		Tourist activity	Tourist Activity Exploring Locations Booking Accommodation Environment Conservation Health Spa and Relaxation Visiting Friends and Relatives	Sightseeing Extreme Sports
Keyword to OQE ratio 4:14			*Keyword to OQE* ratio 4:34		

Table 13 (continued). Expansions for queries Q8 and Q12.

Query 8			Query 12	
Term	*S+S OQE*		Term	*S+S OQE*
Ecotourism	Ecotourism Tourism Industry		Holiday destination	Holiday Destination Tourist Destination Natural Area Resort Foreign Country Abroad Gateway City Beach Resort Ski Resort
Foreign tourist	Foreign Tourist Overseas Tourist International Tourist Tourist		Abroad	Abroad Holiday Destination Tourist Destination
Holiday	Holiday Activity Holiday Package Holiday Sun Sea and Sand Holiday Cruise Holiday Vacation Short Break Fly-Cruise Holiday City Break		Tourism organisation	Tourism Organisation Destination Management Company National Tourist Board Regional Development Agency Tourism Skills Organisation Training and Development Organisation Destination Marketing Organisation Regional Tourist Board
Sightseeing	Sightseeing Tourist Activity		Ski resort	Ski Resort Resort Holiday Destination Tourist Destination
Keyword to OQE ratio 4:17			*Keyword to OQE* ratio 4:17	

3.3.4 Summary of *OQE* Query Search Options

For convenience, the summary of T401, T416 and T438 *optional* and *must-have* query term search options, based on keyword-only and *All*, *S+S* and *S+S+R OQE*s (subsection 3.1.6), are shown again in Table 14. Note, in T401, Ko vs. Oo and Km vs. Om were based on 6 queries using *All OQE* mode and 4 queries based on *S+S OQE* mode.

Table 14. *OQE* query mode matrix used with the TREC topics.

	Ko vs. Oo	Km vs. Om	Ko vs. Oo vs. Oro	Km vs. Om vs. Orm
T401 Immigration	*All OQE, S+S OQE*		n/a	
T416 Three Gorges Project	*S+S OQE*		*S+S+R OQE*	
T438 Tourism	*S+S OQE*		n/a	

The traversal choices for the three experiments were made for several reasons.

i. T401 – the Immigration ontology context was specifically developed as a contextually relevant ontology for the query topic. However, as it is a relatively flat hierarchy, it was decided to compensate for potential *OQE* limitations by conducting some queries using *All OQE,* i.e. the whole ontology hierarchy, and some based on *S+S OQE*.

ii. T416 – the Hydro-electric ontology context was a similarly bespoke and flat hierarchy. The decision to conduct both *S+S OQE* and *S+S+R OQE* comparison modes provided and alternative way of maximising *OQE* opportunities, compared to the T401 approach.

iii. T438 – as the larger Tourism ontology was constructed before the TREC query topics were selected, and had a relatively deeper class hierarchy, it offered more opportunity to rely solely on *S+S OQE*.

4 RESULTS

This chapter firstly provides a factual presentation of the key statistics for each of the three TREC retrieval experiments, together with selected query search effectiveness P&R graphs, and summary radar graphs, showing query mode success comparisons at 10%, 20% and 30% recall points. Each experiment also includes an interpretation of the overall P&R results, in terms of search effectiveness, and the assessments were enabled by TREC's support of a set of query relevance judgements for each query topic document pool.

Summary statistics are shown in Table 15. The query matrices resulted in approximately 18 million document interrogations, based on 180 queries to achieve query mode comparisons. Total queries are the product of the number of queries and the (comparison) query term search options, e.g. the T401 query group Q1-Q10 comprises a query set of 4 query term search options (Ko, Oo, Km and Om) for each query, resulting in 40 queries in total.

Table 15. Summary statistics of TREC folder and document queries executed.

TREC topic	TREC folders queried (folder group)	Number of relevant documents	Total queries (query x search options)	Total number of documents in TREC folders	Total document reads (queries x documents)
T401 Immigration	80 (WT01-WT02)	37	40 (10 x 4)	13,065	522,600 (40 x 13,065)
T416 Three Gorges Project	760 (WT01-WT19)	10	60 (10 x 6)	160,838	9,650,280 (60 x 160,838)
T438 Tourism	480 (WT01-WT12)	36	80 (20 x 4)	96,885	7,750,800 (80 x 96,885)
Totals	1320	83	180		17,923,680

As discussed in section 3.3, the document search was truncated at optimal points to minimise the processing time; this resulted in 83 relevant documents being targeted across the three query topics. The density of the 83 relevant documents in the overall, truncated document set (WT01-WT19 contained 160,838 documents) was approximately 0.05% (83/160,838), compared to 0.92% density for the complete, non-truncated WT2g test collection of 50 query topics, i.e. a pool of 2,279 relevant documents in the 247,491 document collection (folders WT01-WT28).

The graphs in the three main experiments present the results for both single and grouped query set P&R measures. Where grouped queries are presented, precision scores are based on relevant documents counted at each 10% recall point (i.e. a precision percentage is calculated based on the average *number* of relevant documents returned). Where separately stated, group precision scores are also presented using a method of *pooling* the P&R curves (van Rijsbergen, 1979) in a macro-evaluation averaging (MEA) technique; this gives a precision percentage based on the average of precision *percentages*. The two *group* P&R comparison measures were used to provide an alternative view between measures, e.g. where document volumes in one query may distort number averaging results – see later T416 comparison between Fig. 93 and Fig. 94.

To validate the use of a truncated document set, a test set of comparison queries were conducted, using all 247,491 documents in the TREC corpus; the P&R measures of queries on the full corpus were then compared to the same queries on the truncated set. The results of an average of the P&R profile percentages (i.e. MEA measure) are shown in Fig. 68 (full corpus) and Fig. 69 (truncated); they are based on a sample of 9 queries executed in Ko, Oo and Oro query modes across the T401, T416 and T438 topics. The comparison graph y-axes for the validation test are based a 0% to 50% precision scale, for better visual understanding.

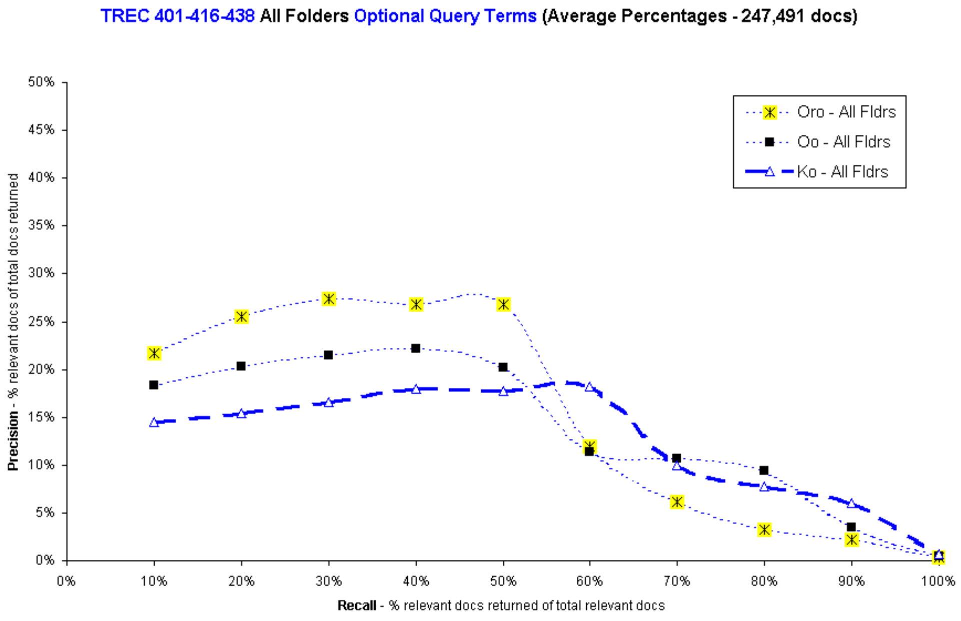


Fig. 68. Validation Test MEA-based P&R results using full TREC corpus.

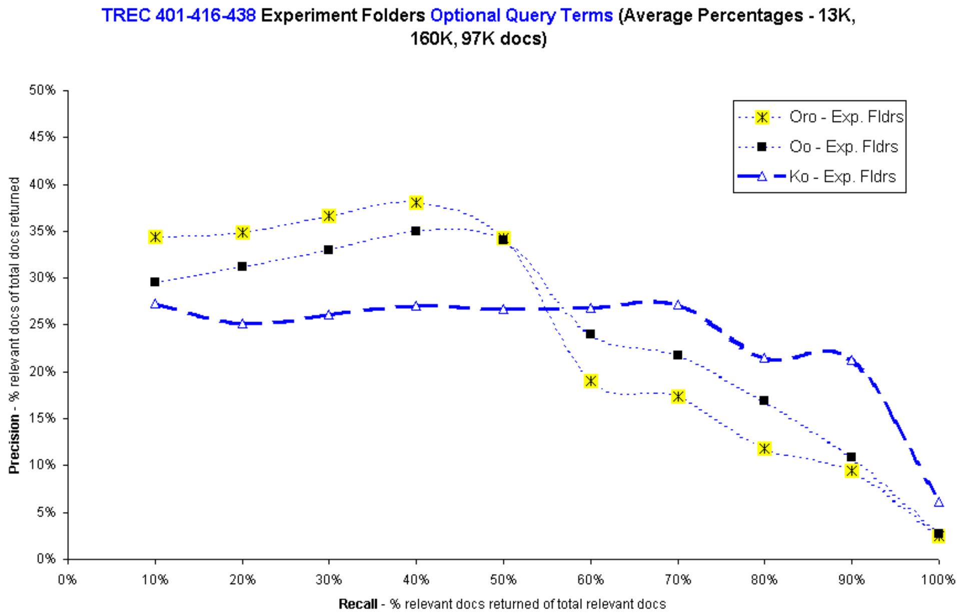


Fig. 69. Validation Test MEA-based P&R results using truncated TREC document sets.

The purpose of the comparison was to see if the *relative* performances of the Ko, Oo and Oro P&R profiles changed markedly when truncated document sets were searched compared to the complete document collection. The results suggest that the two query execution runs produced fairly similar *relative* Ko, Oo and Oro P&R profiles; after taking account that querying a much larger corpus would inevitably produce lower precision percentages. Both runs demonstrated that, between the 10 to 50% recall points, the best precision results were in Oro query term search mode, followed by Oo mode; with the least favourable in Ko mode. This was considered important given that search engine results are ranked by degree of relevance. The P&R profile comparisons, based on unit average percentages, are not shown as the full corpus failed to exceed 4% and the truncated set failed to exceed 8%; the precision curves were much flatter than the MEA-based graphs, with generally minimal difference between Ko, Oo and Oro P&R profiles in both graphs, given the low percentages. On the basis of the validation test results, it was considered justifiable to conduct the T401, T416 and T438 experiments using the representative search pool.

In this results discussion, reference is made to either *typical* or *atypical* P&R profiles; a *typical* profile is one that is representative of a group and an *atypical* profile means not representative of a group. A *typical* P&R profile does not reflect the *average* for the group but is dependent on its frequency in the group (representing the *mode*), e.g. if say 6 of 10 graphs have similar profiles they would be considered *typical*; this distinction will be addressed slightly differently in T438 because of the mix of results generated. It should also be noted that T401 and T416 P&R graph y-axis (precision) scales range from 0% to 100%, whereas T438 graph precision scales range from 0% to either 10% or 30%. The T438 graph y-axis ranges have been reduced for presentation purposes, as many of the maximum precision values returned were below 10%.

The success of each of the TREC *OQE* experiments will be determined by comparing *OQE* P&R curve outcomes against the "base" keyword P&R curve profile, as discussed in subsection 3.1.7. The key determinant of a query mode's search effectiveness will be the precision outcomes in the early, low recall intervals, i.e. it will be based primarily on the average precision value of the 10%, 20% and 30% recall intervals (APV).

The experimental data for all P&R graphs is presented in Appendix F. Graphs not shown in the results section are provided in Appendix G. APV calculations are shown in Appendix I.

4.1 T401 'FOREIGN MINORITIES' EXPERIMENT RESULTS

The T401 document set cut-off, at WT02, provided for 37 of 45 pool relevant documents to be targeted across 13,065 documents. Each of the 10 T401 query sets was conducted in 4 query modes to provide two 2-way comparisons, i.e. Ko vs. Oo and Km vs. Om. As shown in the query matrix in Table 8, subsection 3.3.1, query sets Q1-6 compare keyword queries against *All OQE*, i.e. *OQE* traversal of every class in the ontology, whereas query sets Q7-10 compare keywords against *S+S OQE*, i.e. *OQE* traversal for the base keyword plus sub and super classes. The following P&R graphs demonstrate search effectiveness in T401 based on *optional* and *must-have* query mode comparisons.

4.1.1 Comparing *Optional* Search Mode P&Rs (Ko vs. Oo)

The following two graphs show P&R outcomes for query group Q1-6 for the *optional* query modes: merged P&R results of query group Q1-6 are shown in the graph in Fig. 70, i.e. Ko versus Oo (*All OQE*), and comparison MEA-based P&Rs are shown in Fig. 71.

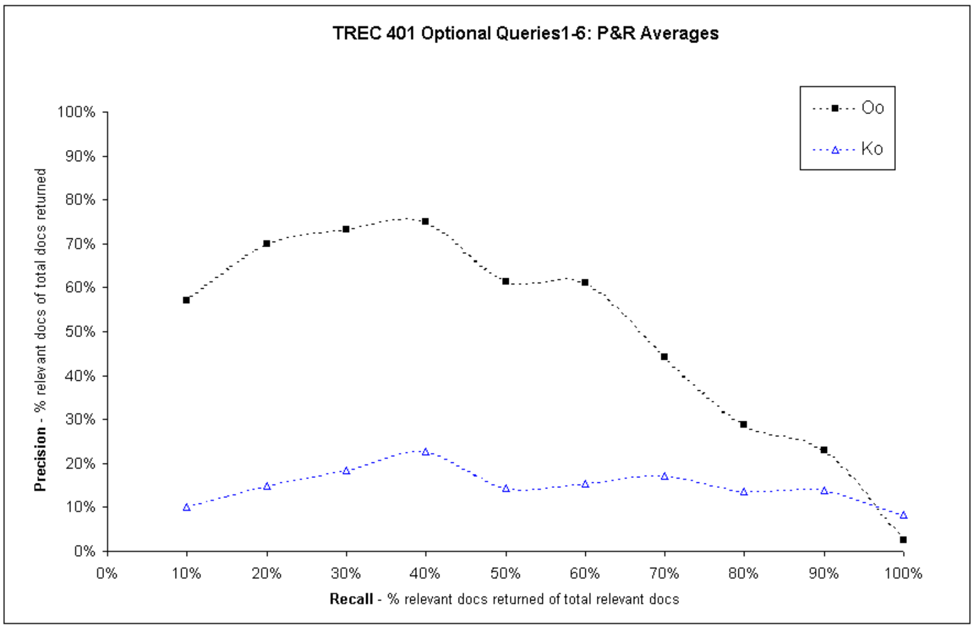


Fig. 70. T401 P&R for *optional* queries Q1-6.

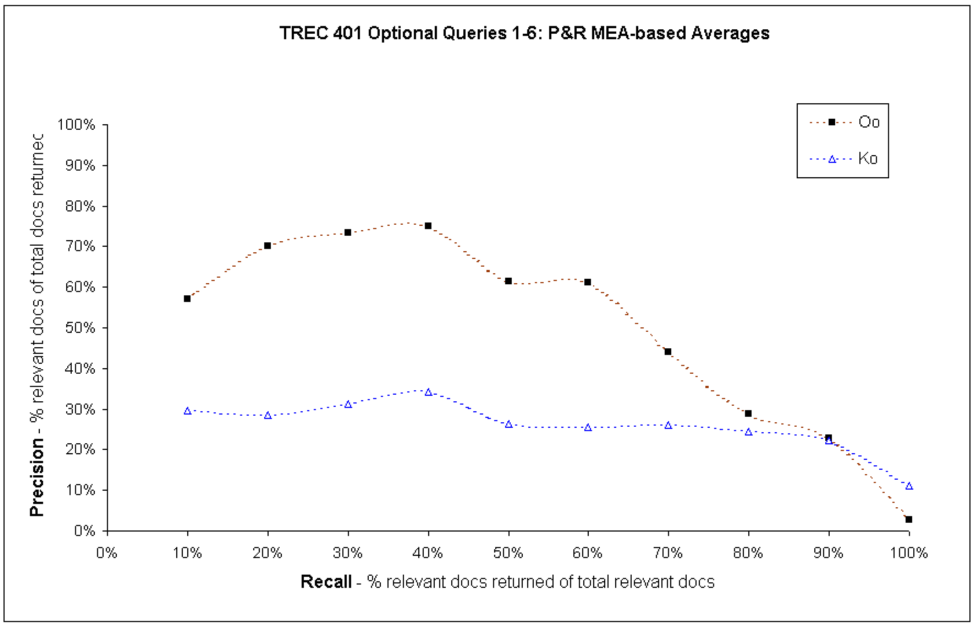


Fig. 71. T401 P&R for *optional* queries Q1-6 - MEA measure.

Primary observations are that the Oo (*All OQE*) curve shows a strong precision performance over Ko in both graphs: Fig. 70 shows that the APV for Ko was 14% and Oo was 67%. Fig. 71 shows that the MEA-based APVs were 30% for Ko and 67% for Oo. Therefore, both graphs show that *OQE* search effectiveness was very good compared to keyword only. The secondary outcomes are that Oo mode achieved a strong precision performance over Ko across the 10-90% recall points and recall was 100% for both Ko and Oo query modes. There were no marked differences in returned document numbers across the queries in the group.

The next two graphs show individual P&Rs for query sets Q4 and Q6. The Ko curve for the individual query sets Q1, 2, 3, and 5 demonstrated *typical* results, i.e. they were consistent with the group Fig. 70's Ko curve and are therefore not shown; whereas *atypical* results were found in query sets Q4 (the keyword *OQE* for Q4 is shown in Table 9, subsection 3.3.1) and Q6, as represented by Fig. 72 and Fig. 73 respectively.

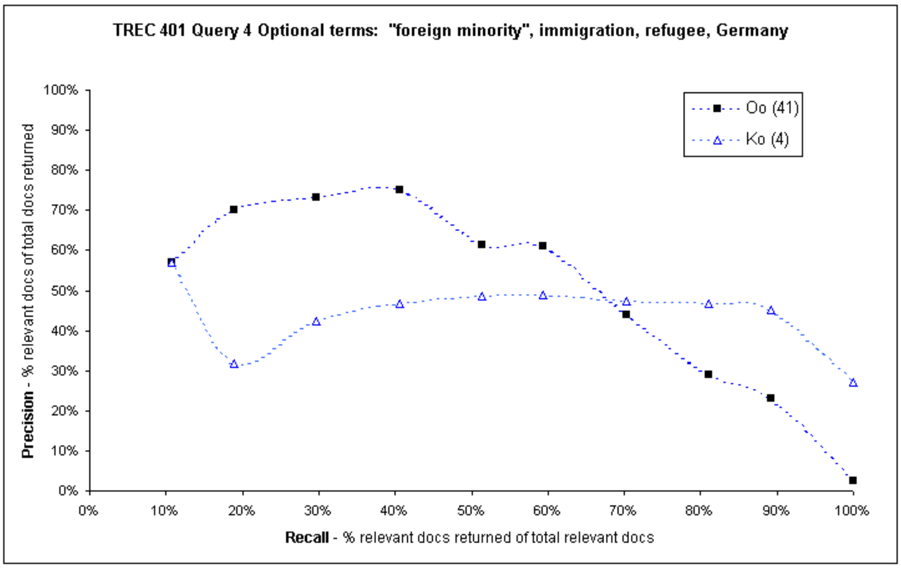


Fig. 72. T401 P&R for *optional* query Q4.

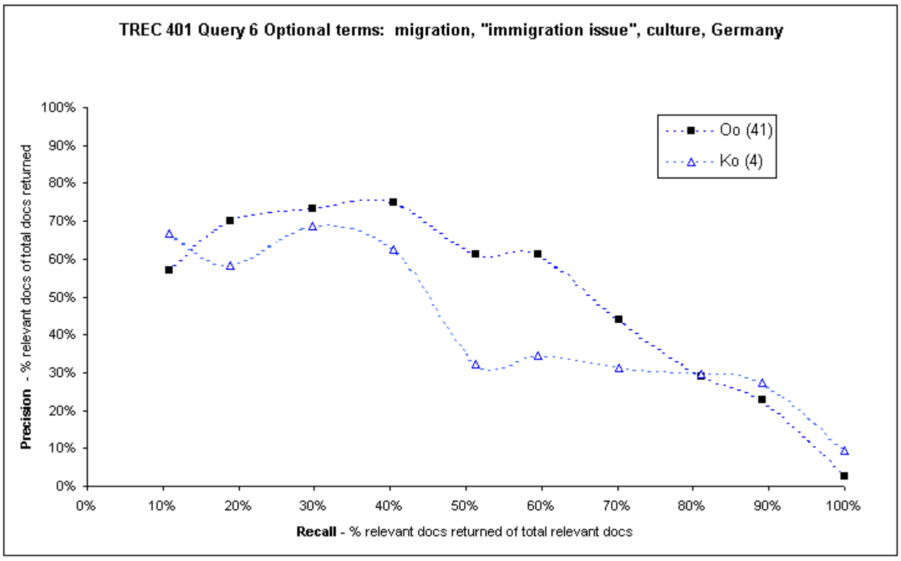


Fig. 73. T401 P&R for *optional* query Q6.

Whilst variable precision results can be seen in both Q4 and Q6, the primary considerations are that in Fig. 72, Q4's APV for Ko was 44% and, for Oo, was 67%; this demonstrated a good Oo (*All OQE*) precision performance over Ko. In contrast, Fig. 73's Q6 APVs were 65% for Ko and 67% for Oo; showing that negligible improvement was achieved using *OQE*. Therefore, mixed results were evident using the primary APV measure. However, a secondary observation is that, whilst Ko mode was competitive with Oo in Q4 and slightly better than Oo in Q6 (at 10% recall), up to 70% recall, Oo mode still produced better overall precision overall in both Q4 and Q6.

The next six graphs show results for query sets Q7-10, based on Ko versus Oo (*S+S OQE*). The merged result of query group Q7-10 is shown in Fig. 74. In this query group, there is little difference between the Oo and Ko P&R profiles, with Ko precision marginally better than Oo at 10% and 20% recall but lower at 30% to 80% recall.

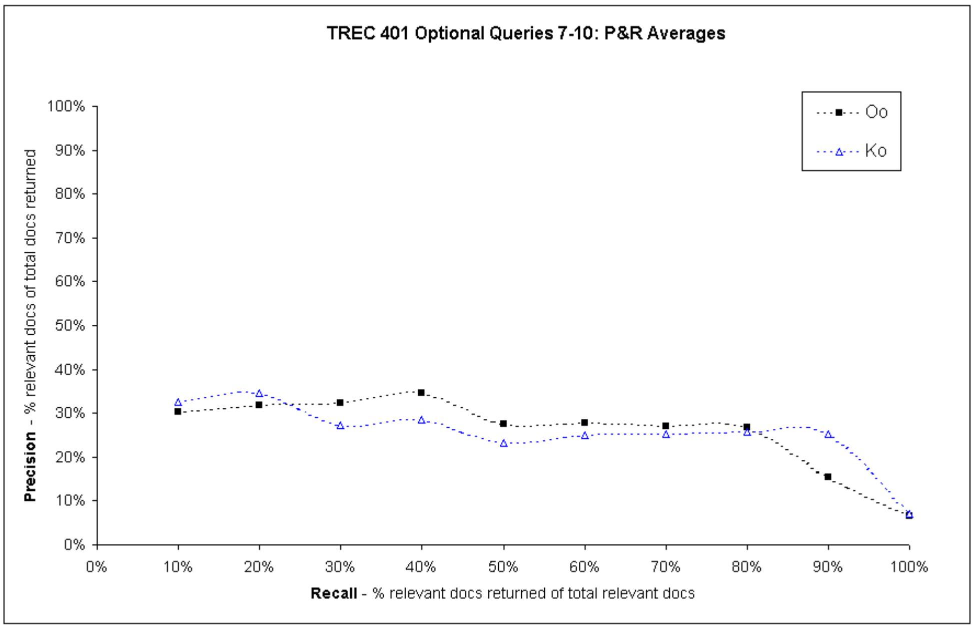


Fig. 74. T401 P&R for *optional* queries Q7-10.

Fig. 74's primary APV measures are 32% for Ko and 31% for Oo, showing that *OQE* search effectiveness was poor, with negligible difference between modes. However, the group results are somewhat distorted, given that three of the four query sets Q7, 8 and 9 all showed better Oo mode precision results than Ko; the group results were affected solely by markedly higher than normal document numbers returned in query set Q10. The MEA comparison measure shown later in Fig. 79 minimises the effect of Q10. However, given the mix of results, the individual P&R curves for Q7, 8, 9 and 10 are shown in Fig. 75, Fig. 76, Fig. 77 and Fig. 78, respectively. The query expansion terms for Q8 and Q10 were shown in Table 9, subsection 3.3.1.

Fig. 75's primary APV measures for Q7 are 53% for Ko and 61% for Oo, showing that, whilst the difference between Ko and Oo was negligible between 10% and 20% recall, *OQE* achieved a 15% APV improvement over keyword, with Oo precision some 8 percentage points higher.

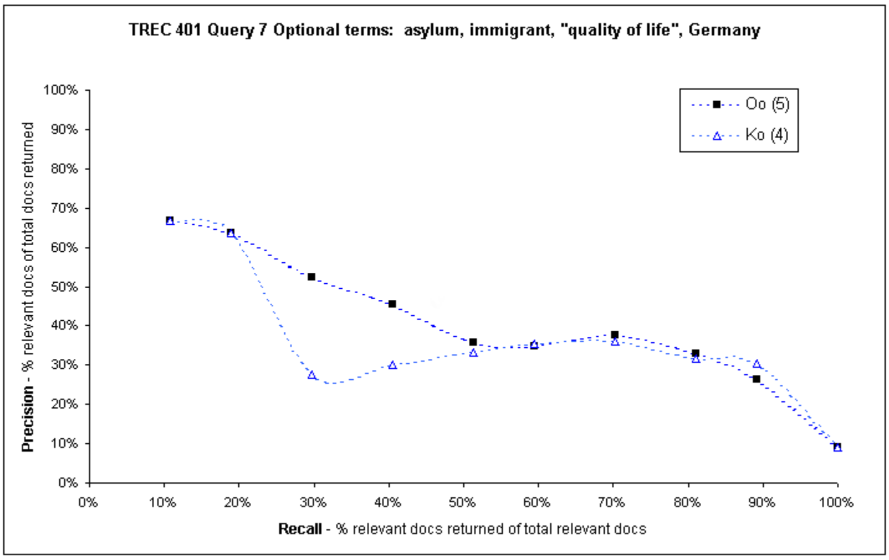


Fig. 75. T401 P&R for *optional* query Q7.

Fig. 76's Q8 primary measures are 23% for Ko and 63% for Oo; showing that APV-based *OQE* search effectiveness was very good compared to keyword only.

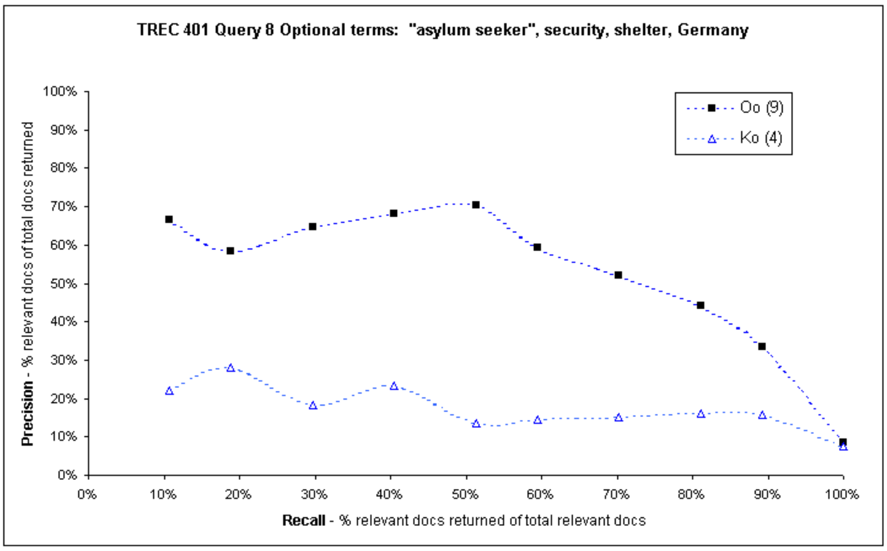


Fig. 76. T401 P&R for *optional* query Q8.

A similarly good *OQE* improvement in search effectiveness was evident in Q9, Fig. 77, with APV measures of 34% for Ko and 59% for Oo. Finally, whilst Fig. 78's APV measures for Q10 show weaker search effectiveness for both modes, with 13% for Ko and 33% for Oo, *OQE* resulted in a good Oo performance compared to keyword only.

The MEA-based Q7-10 group comparison is shown in Fig. 79; it has produced a more representative P&R curve than the earlier Fig. 74, with the impact of the markedly higher number of Q10 documents minimised. The primary APV measures are 35% for Ko and 49% for Oo; showing that *OQE* search effectiveness was very good compared to keyword only.

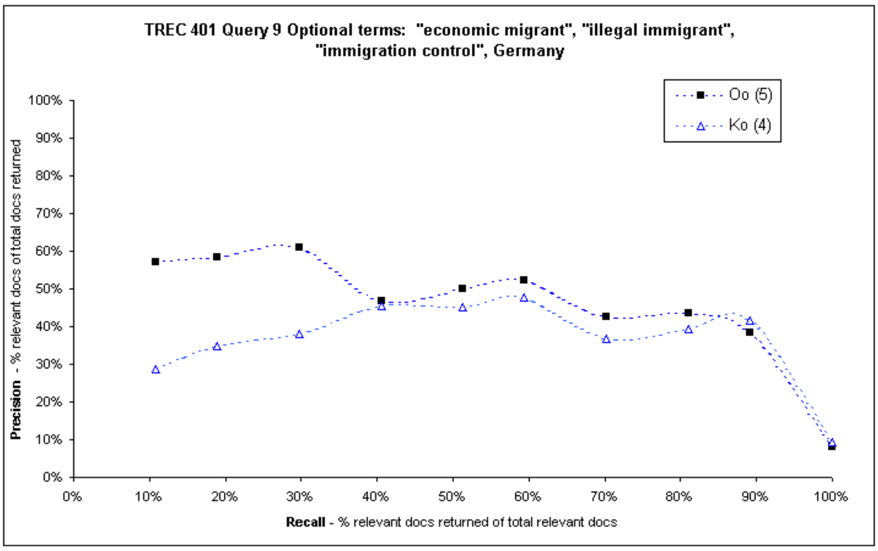


Fig. 77. T401 P&R for *optional* query Q9.

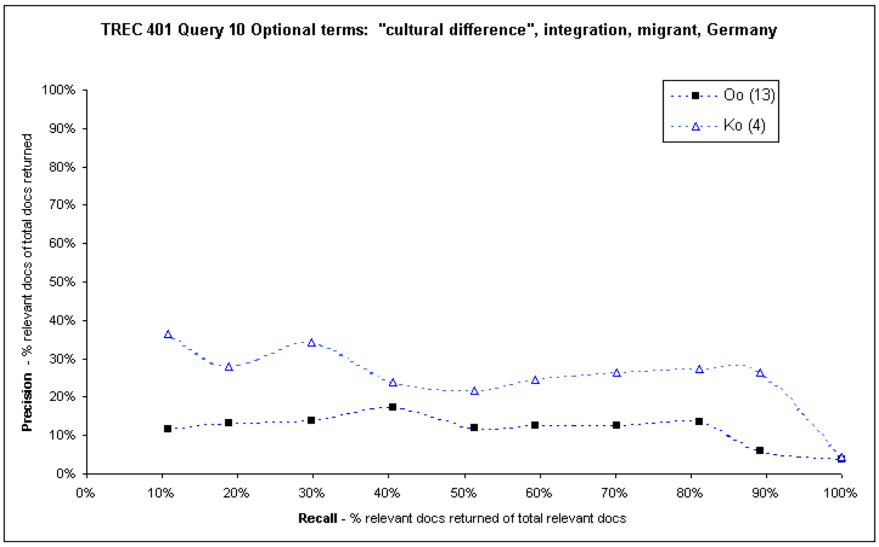


Fig. 78. T401 P&R for *optional* query Q10.

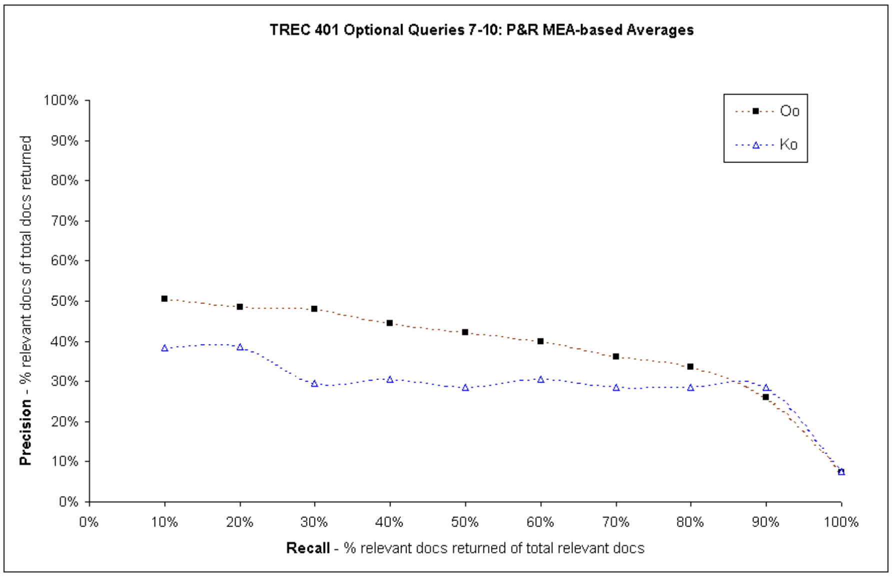


Fig. 79. T401 P&R for *optional* queries Q7-10 - MEA measure.

4.1.2 Comparing *Must-have* Search Mode P&Rs (Km vs. Om)

The next two graphs show the *must-have* results of the merged queries, Q1-6 Km versus Om (*All OQE*); Fig. 80 provides a comparison with the MEA-based P&R shown in Fig. 81. In both measures the Om shows a better precision result over Km at all recall points; they are similar to the *optional* mode profiles in Fig. 70 and Fig. 71. Individual query profiles (not shown) were similar to their *optional* mode counterparts, with Q4 and Q6 profiles also *atypical*.

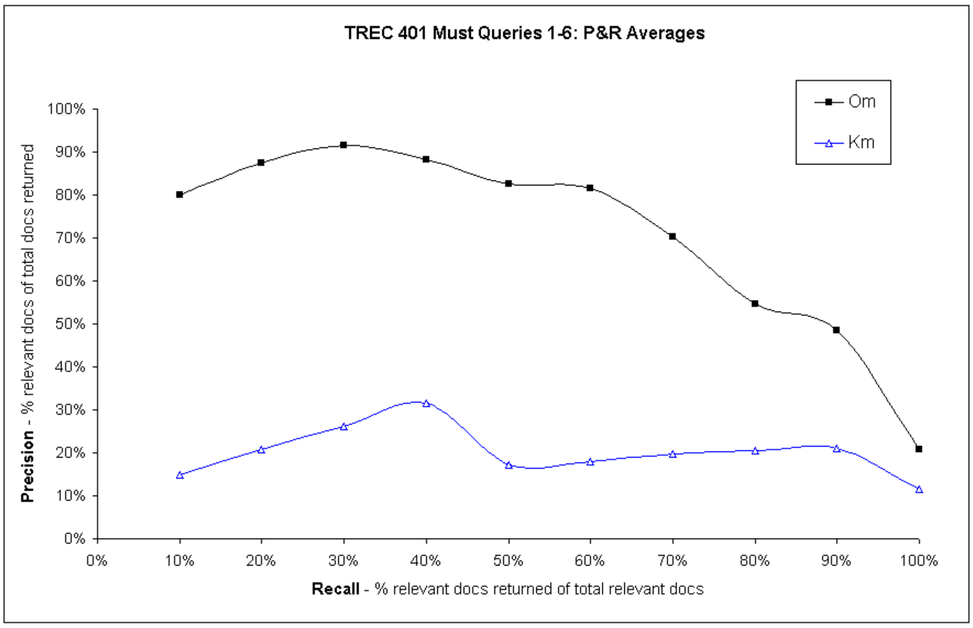


Fig. 80. T401 P&R for *must-have* queries Q1-6.

Fig. 80's APV measures are 21% for Km and 86% for Om; giving a very good *OQE* search effectiveness performance compared to keyword. The Fig. 81 MEA-based APVs, 37% for Km and 86% for Om, also confirm the very good *OQE* search effectiveness result over keyword.

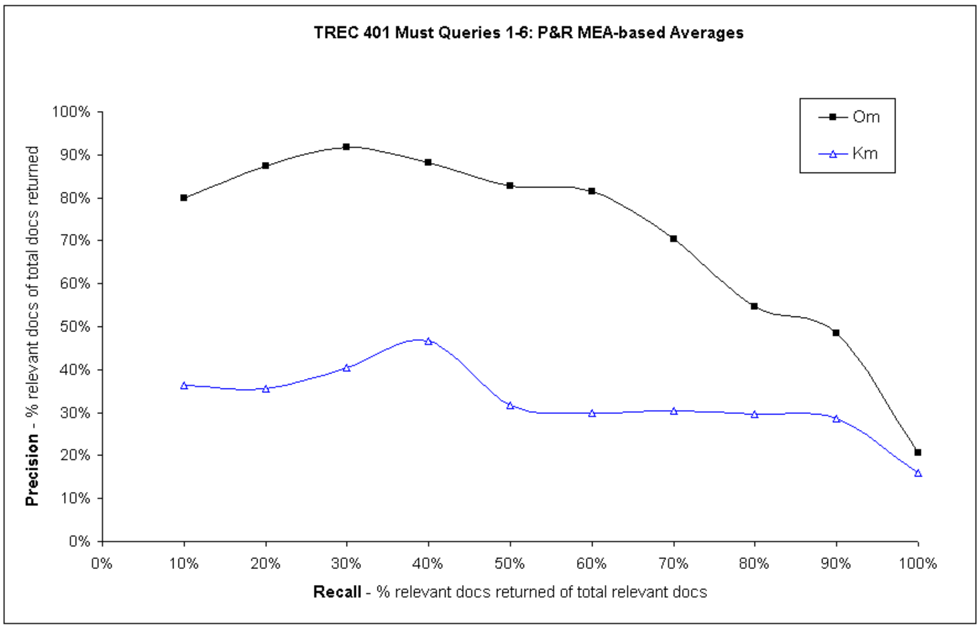


Fig. 81. T401 P&R for *must-have* queries Q1-6 - MEA measure.

The merged results of *must-have* query group Q7-10, Km versus Om (*S+S OQE*), are shown in Fig. 82, with the MEA-based version in Fig. 83. As both P&R profiles were consistent with their *optional* mode counterparts (Fig. 74 and Fig. 79), individual Q7-10 graphs are not shown.

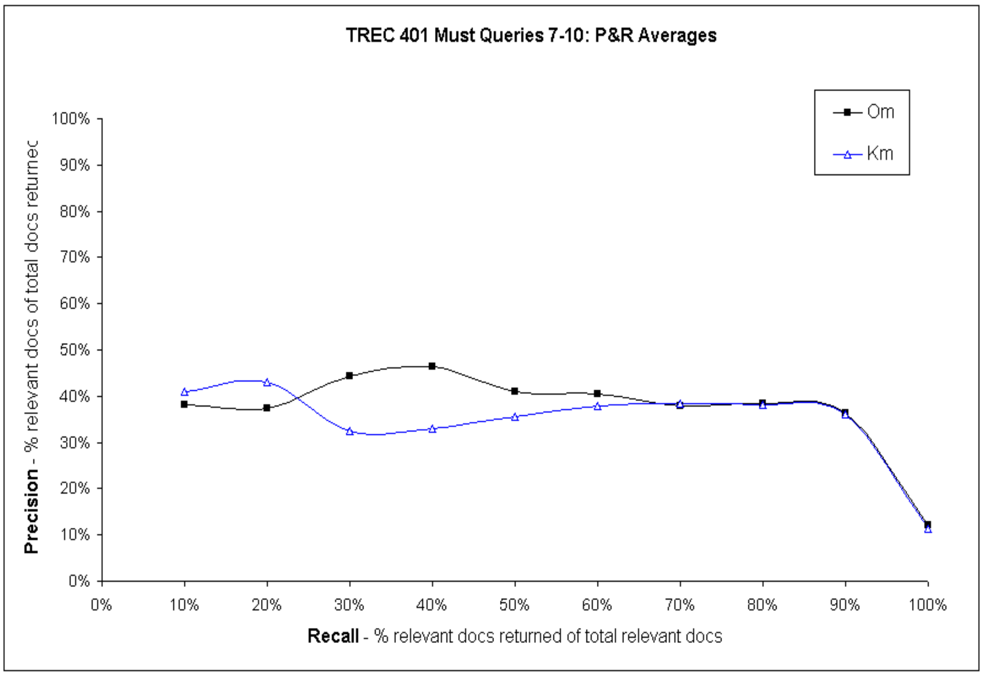


Fig. 82. T401 P&R for *must-have* queries Q7-10.

Fig. 82's APV measures are 39% for Km and 40% for Om; this shows that *OQE* search improvement was poor, with negligible difference between modes. The Fig. 83 MEA-based APVs, 42% for Km and 55% for Om, demonstrated a good *OQE* search effectiveness performance, given the 25% improvement in APV level.

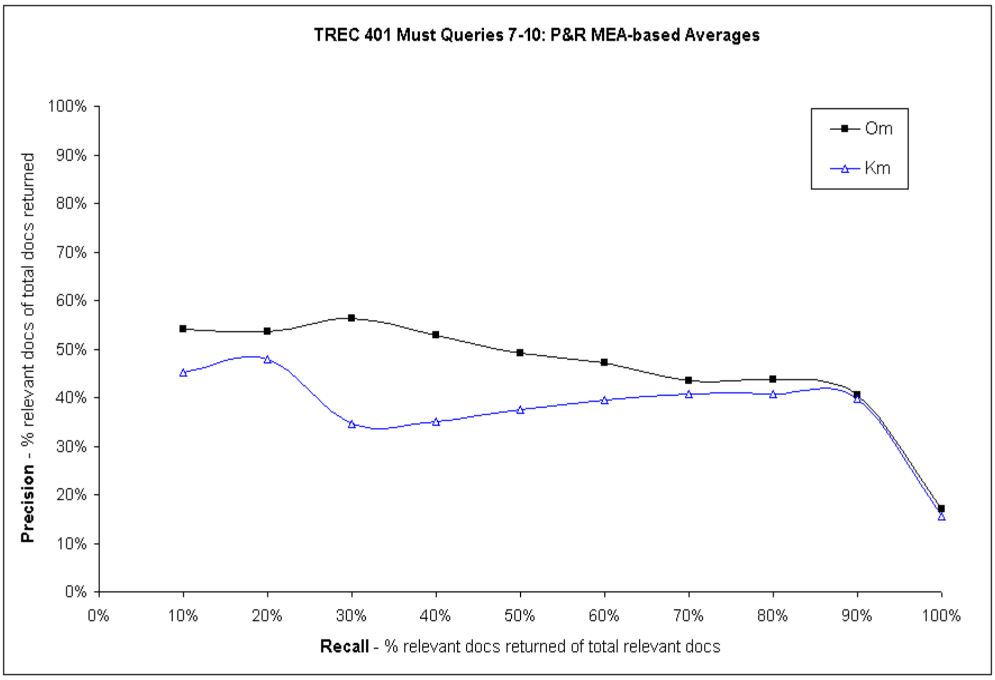


Fig. 83. T401 P&R for *must-have* queries Q7-10 - MEA measure.

4.1.3 Overall Group Query Term Search Mode P&Rs

This final results subsection shows the combined T401 P&R results for all 10 queries, first in *optional* query term search mode (Ko vs. Oo) and then in *must-have* modes (Km vs. Om); in effect these summarise the T401 experiment. The *optional* mode graph is shown in Fig. 84, with the comparison MEA-based graph shown in Fig. 85.

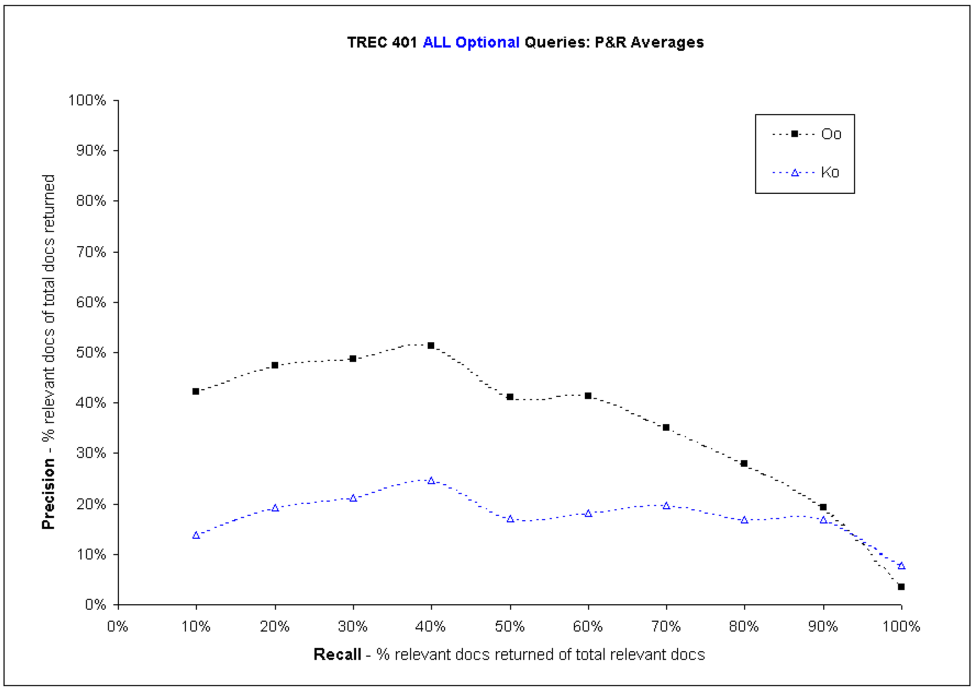


Fig. 84. T401 overall P&R for *optional* queries.

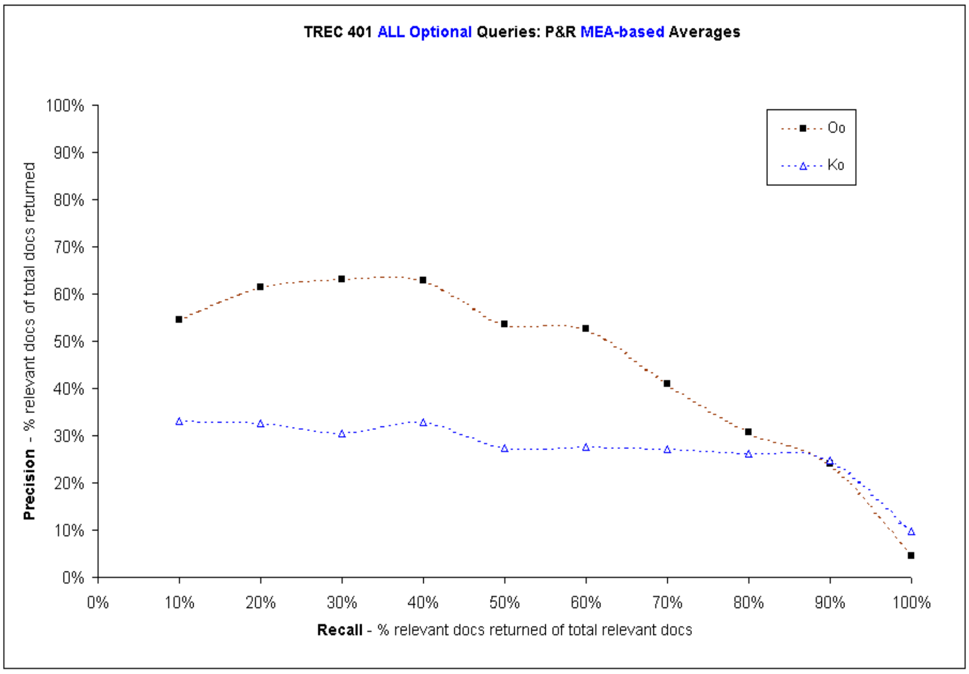


Fig. 85. T401 overall P&R for *optional* queries - MEA measure.

Primary observations are that the Oo curve shows a strong precision performance over Ko in both graphs: Fig. 84 shows that the APV for Ko was 18% and, for Oo, was 46%. Fig. 85 shows that the MEA-based APVs were 32% for Ko and 60% for Oo. Therefore, both graphs show that *OQE* achieved a very good improvement in search effectiveness compared to keyword only. A secondary outcome is that both Fig. 84 and Fig. 85 show *OQE* resulted in much improved precision between the 10-80% recall points, with the MEA-based graph showing raised precision levels overall.

The group *must-have* results are shown below in Fig. 86 (Km vs. Om), with the comparison shown MEA graph in Fig. 87; they demonstrate similar result comparisons.

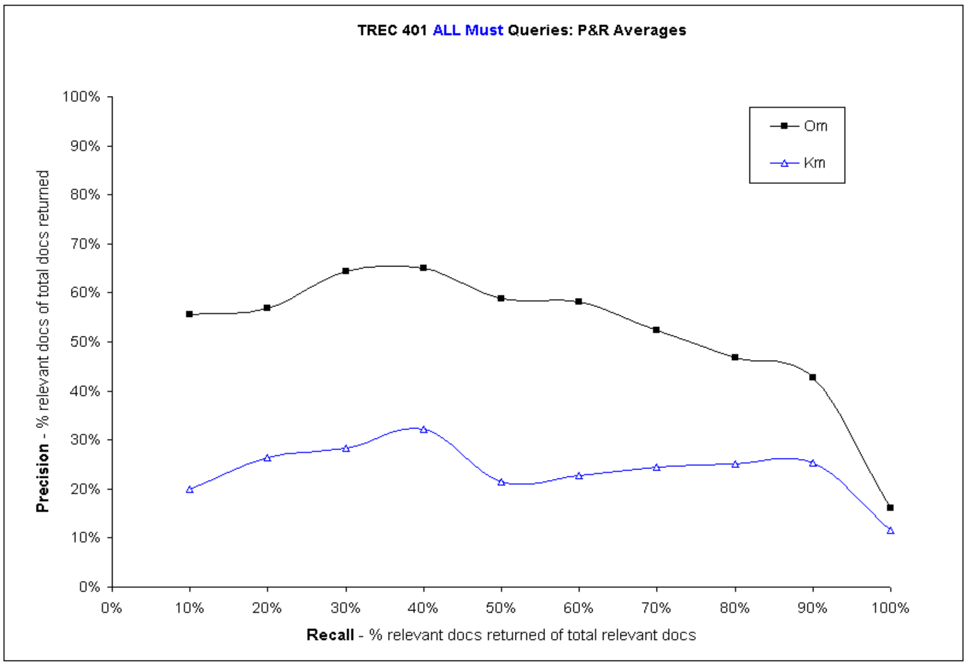


Fig. 86. T401 overall P&R for *must-have* queries.

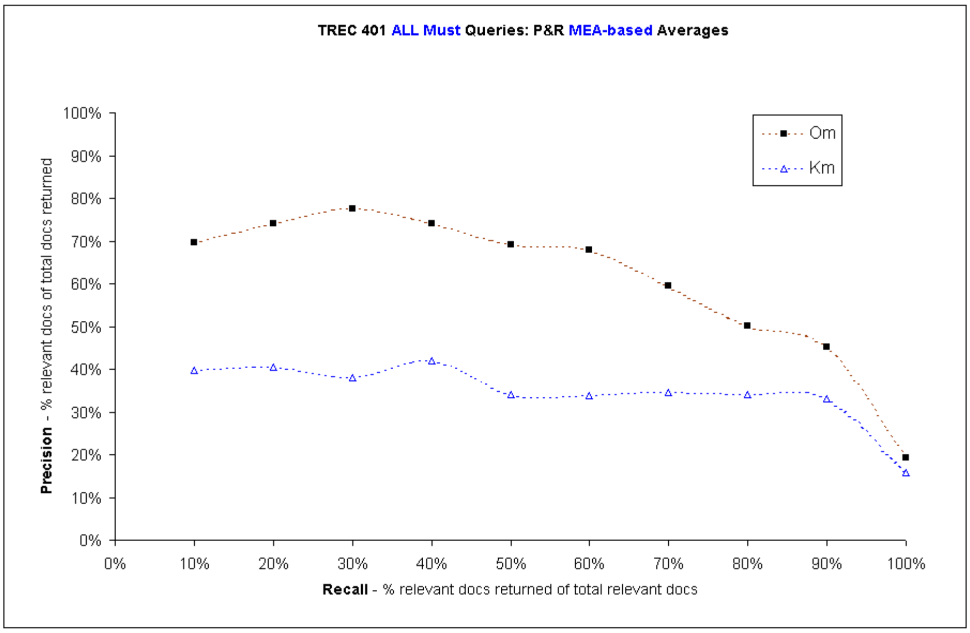


Fig. 87. T401 overall P&R for *must-have* queries - MEA measure.

Primary outcomes are that the Om curve shows a strong precision performance over Km in both graphs: Fig. 86 shows that the APV for Km was 25% and, for Om, was 59%. Fig. 87 shows that the MEA-based APVs were 39% for Km and 74% for Om. Again, both graphs demonstrate good *OQE* search effectiveness compared to keyword. The secondary outcome is that, again, both graphs show markedly better *OQE* precision over keyword, this time extended to 90% recall; although Fig. 87's MEA-based graph showed higher precision (between +15% and +20%) across 10-70% recall points for both keyword and *OQE*.

General Observations:

The density of the 37 relevant documents in the T401 document set (WT01-WT02) was approximately 0.28% (37/13065) compared to an average 0.05% for the overall document set for the 3 experiments (WT01-WT19).

In the Q1-6 query group, *All OQE* traversal generated an average 41 query terms, of which 37 (90%) were matched in documents; this resulted in a keyword to *OQE* ratio of 4:41. In the Q7-10 query group, the *S+S OQE* traversal generated on average only 8 query terms, all (100%) of which were matched in the document search, giving a 4:8 keyword to *OQE* ratio. For the full group of 10 query sets, the average *OQE* ratio was 4:28, with 25 classes (89%) matched in documents. An analysis of *OQE* term matches is shown in Appendix E.

The inconclusive P&R results of group Q7-10 query sets (Fig. 74 and Fig. 82) may be a consequence of the group's low 4:8 *OQE* ratio, e.g. when compared to the markedly higher Q1-6 group *OQE* ratio of 4:41. The Q1-6 *optional* and *must-have* P&Rs (Fig. 70 and Fig. 80 respectively) conclusively favour *OQE*.

It is clear that the merged P&R results for queries Q1-6 *All OQE* produced a markedly better *OQE* precision result than the keyword only result, in both *optional* and *must-have* query term searches. The average *OQE* result of 89% of terms matched in documents, suggests that the ontology context, developed independently from the documents, was very relevant for the query experiment. All 37 relevant documents were found using both keyword and *OQE optional* and *must-have* query term search options, resulting in 100% recall.

4.1.4 Comparison of Precision Results Across All Query Modes

To provide a different way of demonstrating IR search effectiveness, a league table of precision scores was developed, for each query set, comparing all query modes at recall points 10-30%. Scoring was based on 1-4 points, e.g. if Q1 resulted in Ko 30%, Oo 50%, Km 60% and Om 40% precisions at 10% recall, the modes would be awarded 1, 3, 4 and 2 points respectively. This was repeated for all query sets in the query group and the average score for each mode was calculated as a percentage of maximum available 4 points.

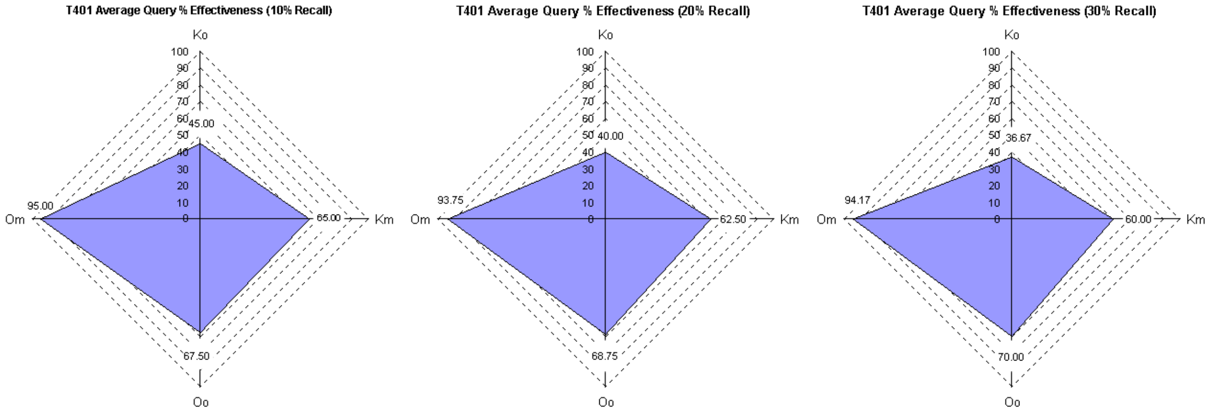


Fig. 88. T401 average query percentage effectiveness.

Fig. 88 shows the search effectiveness comparisons plotted in radar graph for 10%, 20% and 30% recall points. The performance profiles of Ko and Km modes show reducing search precision effectiveness moving from 10% to 30% recall points, with slightly improved Oo and stable Om. At 20% and 30% recall points it is clear that Om and Oo *OQE* produce consistently higher precisions than Km and Ko.

4.1.5 APV Measures

The key determinant of a query mode's search effectiveness is the APV measure and specific individual query set outcomes have been presented in subsections 4.1.1 and 4.1.2. The APVs for all 20 individual query sets, i.e. 10 query sets in both *optional* (Ko/Oo) and *must-have* (Km/Om) modes, are shown in Appendix I. Table 16 provides a summary of the times a query mode APV was either most successful (Top) or performed the same (Tied).

Table 16. Comparisons of T401 query mode APV successes.

Optional Mode			
% Ko Top	% Oo Top		% Tied
10%	90%		0%

Must-have Mode			
% Km Top	% Om Top		% Tied
10%	90%		0%

The results show that, in both modes, *OQE* proved the most search effective in 90% of queries, with keyword 10% and no results tied. This was a very good *OQE* outcome.

4.1.6 Comparing *Optional* and *Must-have* Query Mode Successes

In comparison to Table 16 above, Table 17 shows the number of times each *optional* and *must-have* search mode produced the highest precision at 10, 20 and 30% recall points in the 10 query sets. Tied precision outcomes are shown separately.

On average *optional* Oo query term search was the most effective 77% of the time, with *optional* Ko better only 13% of the time. The two search options tied 10% of the time. In *must-have* mode, Om was the best 83% of the time versus Km 13%, with tied at 3%. At the specific 30% recall point, i.e. equivalent to returning the highest scoring 11 of 37 relevant documents, Oo and Om achieved the most effective result 90% of the time.

Overall, the results support the strong *OQE* APV outcomes shown in subsection 4.1.5.

Table 17. Comparisons of T401 query mode successes.

Optional Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Ko	2	1	1	1.3	13%
Oo	6	8	9	7.7	77%
Tied Results	2	1	0	1.0	10%

Must-have Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Km	1	2	1	1.3	13%
Om	8	8	9	8.3	83%
Tied Results:	1	0	0	0.3	3%

The above results are illustrated in Fig. 89, with tied results shown as Ko~Oo or Km~Om.

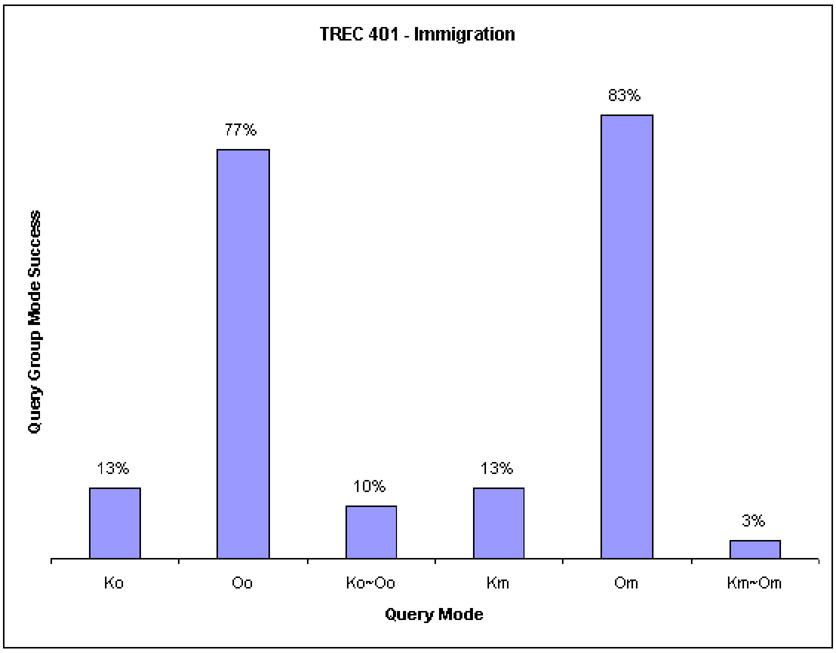


Fig. 89. T401 query mode successes.

4.1.7 Critical Review of Experiment

The Immigration context has a fairly shallow class hierarchy supporting 41 query terms, which limits *S+S OQE* potential; this was the justification for using *All OQE* mode in query sets Q1-6.

The experiments were successful and demonstrated strong *OQE*-based search effectiveness improvement: the APV summary (subsection 4.1.5) shows that, in both modes, *OQE* proved the most search effective in 90% of queries; Appendix I will show the split was 60% *All OQE* and 30% *S+S OQE*, despite keyword mode achieving some competitive results. This was a very good *OQE* outcome, with the most successful APVs obtained using *All OQE*; the learning from this must be that the greater the ontology query expansion (i.e. beyond *S+S OQE*), the better the APV precision measure is likely to be. Wider *OQE* will be assessed further in T416, where a similarly flat ontology hierarchy (hence limited S+S *OQE* potential) will be interrogated by additionally using an *S+S+R OQE* approach, as opposed to *All OQE*.

On reflection, the decision to apply a "direct" weighting (1.0) for all classes, when using *All OQE*, could be criticised; the values would have had an impact on the *tf-idf* algorithm generated rankings, although *All OQE* was not used in 40% of query sets (Q7-10). However, the "direct" weighting effect would be difficult to quantify, as the spread and frequency of terms found in documents would have to be manually checked.

The choice of "Germany" as a query term offered a focused and influential use of a *must-have* query constraint. As will be seen later, in the T416 and T438 experiments, unless an accurate *must-have* mode query term is used (i.e. the *must-have* term is dependent on a precise query objective, i.e. in the TREC statement), the constraint can exclude potentially relevant documents. However, in this experiment the recall was unaffected, as 100% of the 37 relevant documents were found in both the *optional* and *must-have* query term search mode comparisons (Ko vs. Oo and Km vs. Om modes); this demonstrated no increased recall benefit was derived from *OQE* modes.

On reflection, if more time had been available to give a wider base for comparison, it would have been worthwhile conducting all 10 query sets in both *All OQE* and *S+S OQE* versus keyword-only, or using on *S+S+R OQE*. Nevertheless, the use of *All OQE*, to achieve a greater *OQE* ratio, has demonstrated sufficiently clear precision improvement. Finally, the *OQE* results may have been improved if different weightings had been applied, as indicated by the later T401 *S+S OQE* weight reversal experiments, in subsection 4.4.3.

4.1.8 Reflections on Hypotheses

Comments are now provided for the hypotheses proposed in subsection 1.9.2.

Hypothesis (i) – "hierarchical (*S+S*) *OQE* can have a positive impact on precision and recall, although class hierarchy expansions alone may not produce optimal results. Query term-matched classes may have more beneficial wider semantic *relations* with other classes" (*S+S+R OQE*).

The T401 results showed that *OQE* achieved better APV results in 90% of queries; further, the average precision level for both *optional* and *must-have OQE* groups was approximately 30 percentage points higher than keyword-only modes (e.g. 45% vs. 15% in Fig. 84), between the 10% and 60% recall points. Recall performance was not dependent on any particular query mode, as each query mode achieved 100% recall.

The experiment results have provided a strong and positive indication of the search effectiveness benefits of *OQE* and, as *All OQE* accounted for the main impact, the benefits of extending query expansion beyond *S+S OQE*. Whilst *S+S+R OQE* was not employed in this experiment, the APV outcomes support the first part of the hypothesis that "hierarchical *OQE* can have a positive impact on precision".

Hypothesis (ii) - "higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes".

The initial T401 weightings and APV outcomes will be compared with revised T401 weighting experiment outcomes in subsections 4.4.1 to 4.4.3; to provide learning on how term relevance weightings can influence *OQE* precision results.

Hypothesis (iii) – "topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion, despite their potentially restrictive coverage, … as opposed to contextually wider, or more comprehensive, ontologies …. to avoid superfluous query expansion".

The Immigration ontology context was developed specifically for the T401 topic and the average *OQE* term class matching of 89% was the second highest of the three query experiments. The context APV outcomes clearly support the hypothesis, given the positive search effect of *OQE* and good precision improvement over keyword modes.

The benefit of a topic specific, or self-contained ontology context, in maximising contextual relevance and minimising "superfluous" *OQE* (avoiding potentially generalised and less relevant terms), will be further evaluated: by comparing T401's APV and class matching results against T416 and T438, and T401 against an *extended* T401 (see subsection 4.4.4).

4.2 T416 'THREE GORGES PROJECT' EXPERIMENT RESULTS

The T416 document set cut-off at WT19 provided for 10 of 14 pool relevant documents to be targeted across 160,838 documents. Each of the 10 queries was conducted using a set of 6 query term search options to provide two *3-way* comparisons, i.e. Ko vs. Oo vs. Oro and Km vs. Om vs. Orm; therefore the query matrix (Table 10, subsection 3.3.2) was based on each of the 10 query sets comparing *optional* and *must-have* keyword query mode against *optional* and *must-have S+S OQE* and *S+S+R OQE* query modes.

The following provides an assessment of search effectiveness, based on the three query modes, firstly by measuring P&R for the whole query group and then by considering typical and atypical P&R results in individual query sets.

4.2.1 Overall Group Query Term Search Mode P&Rs

This first subsection reviews the search effectiveness results of the combined query group Q1-10 for *optional* then *must-have* query term search options.

Comparing *Optional* Query Modes

The P&R graph comparing the combined results for the *optional* Ko, Oo and Oro modes across the query group Q1-10 is shown in Fig. 90, based on the average *number* of relevant documents returned. The comparison MEA measure, based on the average of *percentages*, is in Fig. 91.

In Fig. 90, the primary outcome is that the APV for Ko was 14%, with 21% for Oo and 27% for Oro; this again demonstrated the benefit of *OQE* and, in particular, *S+S+R OQE*. Therefore, *OQE* provided a good APV improvement in search effectiveness over Ko. A secondary outcome was that both *OQE* modes improved precision to as far as 50% recall.

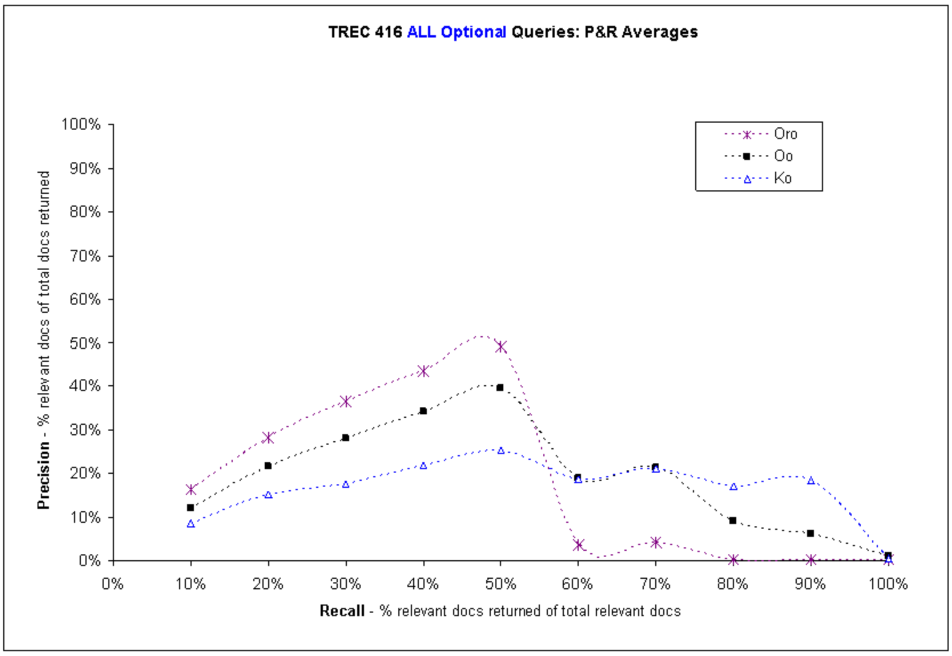


Fig. 90. T416 overall P&R for *optional* queries.

In Fig. 91, the MEA-based primary APV measures were clearer: Ko was 27%, Oo was 35% and Oro was 54%; this particularly confirmed the benefit of *S+S+R OQE*. Therefore, *OQE* provided a very good APV improvement in search effectiveness over Ko. As in Fig. 90, both *OQE* modes improved precision up to 50% recall.

The MEA-based APV measure was much improved for all modes, compared to Fig. 90.

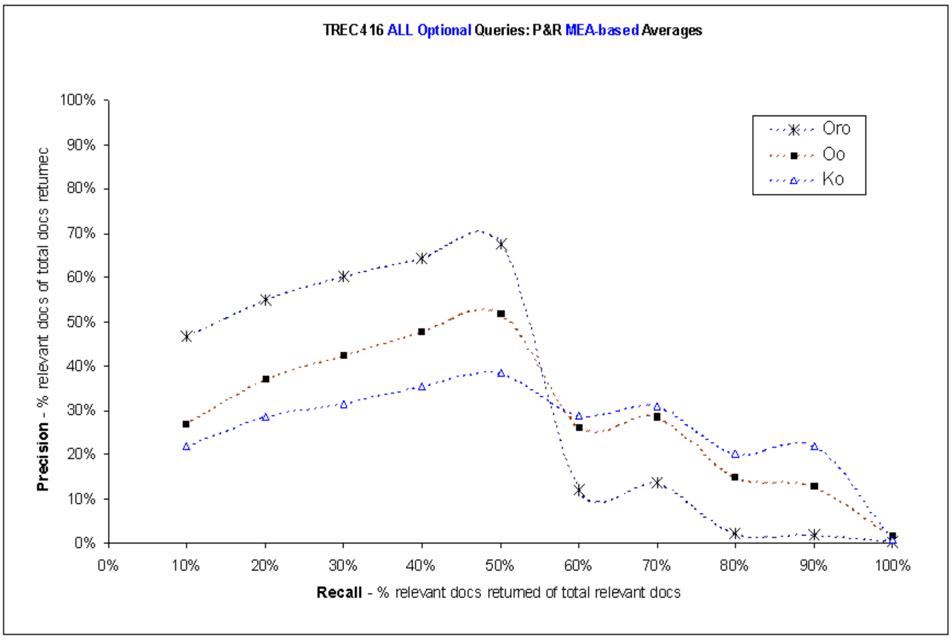


Fig. 91. T416 overall P&R for *optional* queries - MEA measure.

Comparing *Must-have* Query Modes

The combined results for the *must-have* Km, Om and Orm modes across the whole query group are shown in Fig. 92.

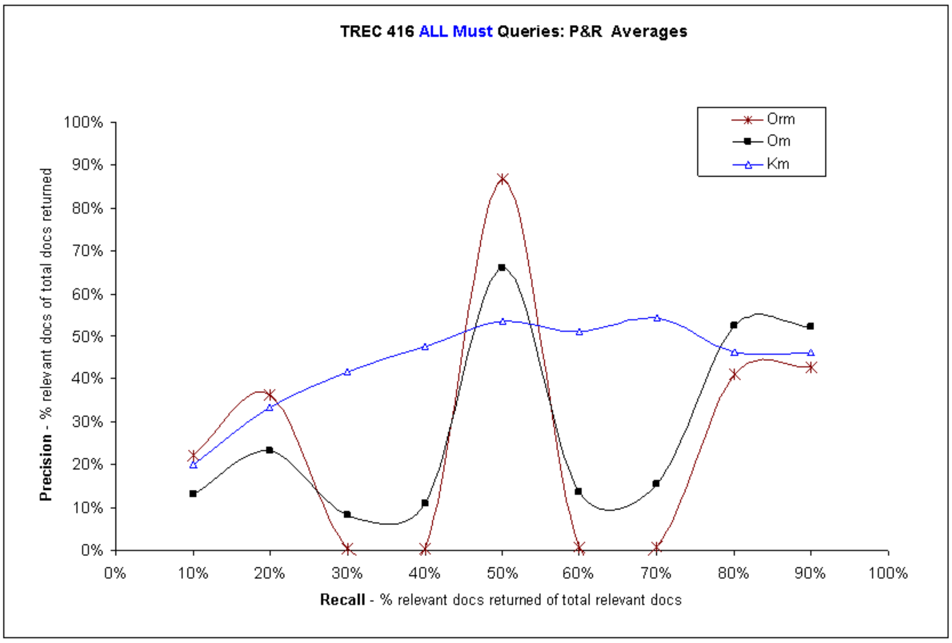


Fig. 92. T416 overall P&R for *must-have* queries.

The primary APV measures are 32% for Km, 15% for Om, and 20% for Orm, giving the impression that *OQE* was poor. However, the graph displays a marked precision aberration in the Orm and Om curves at 30% and 40% recall points; this was caused by a similar situation to the one that affected the T401 group average discussed earlier with Fig. 74. In T416's Fig. 92, the precision performance was affected by abnormal document numbers returned in one query

(Q2) in Om and Orm modes, i.e. 308 (Om) and 6848 (Orm), compared to an average 6 (Om) and 4 (Orm) documents in Qs 1, 3 to 9; as a result, Q2's volumes deflated the group precision. This was validated in the variance-adjusted result shown in Fig. 93, by replacing Q2's 30% and 40% recall figures with the respective average volumes characterised in the other 9 query sets.

Fig. 93's revised group P&Rs resulted in APV measures of 32% for Km, 22% for Om, and 34% for Orm; whilst these showed better *OQE* search effectiveness, Om performance was poor, compared to Km, and Orm provided only negligible improvement over Km.

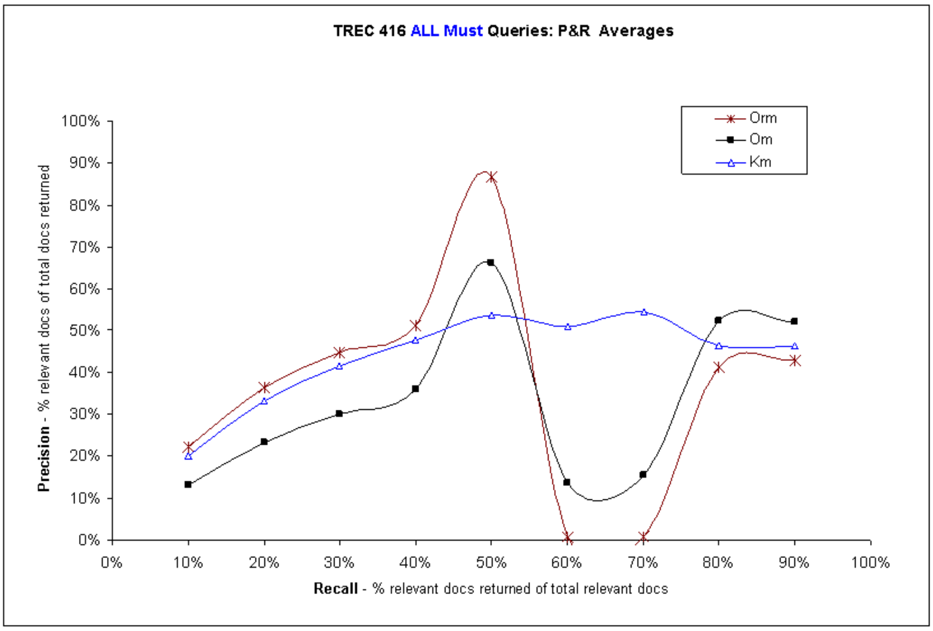


Fig. 93. T416 overall P&R for *must-have* queries – with Q2 revised.

Fig. 93 can now be compared to the (unadjusted) group MEA-based measures in Fig. 94.

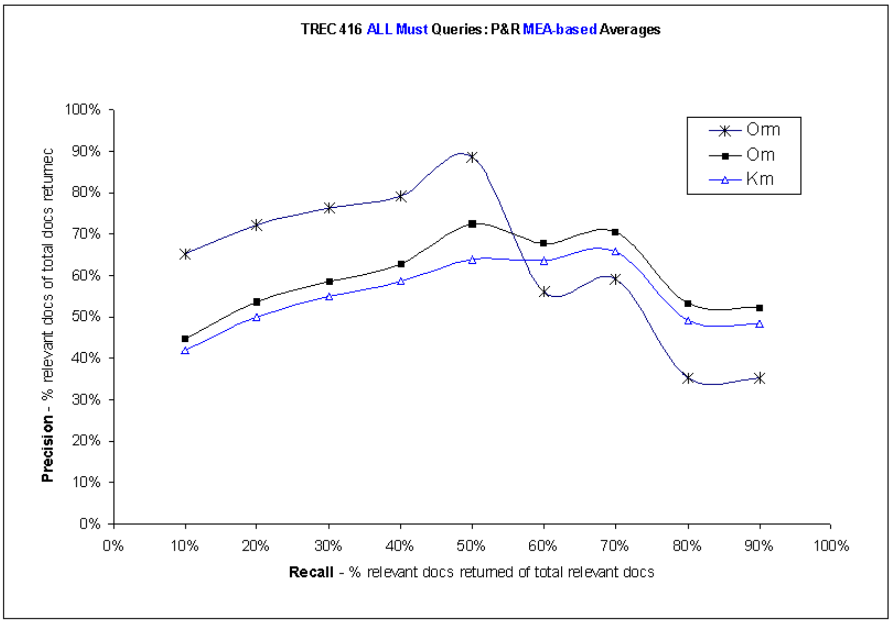


Fig. 94. T416 overall P&R for *must-have* queries - MEA measure.

The MEA measures show better results for all modes, with Orm performance highest but with the Km and Om positions reversed. APV was 49% for Km, 52% for Om, and 71% for Orm. The results demonstrate that *S+S+R OQE* achieved very good search effectiveness. Om performance was poor compared to Km; providing only marginal improvement.

The inconclusive nature of Fig. 93's profile may be accounted for by either, the low 4:8 *OQE* ratio resulting from Om mode, or the Q2 query set figures affecting the query group as a whole; again justifying the use of the MEA comparison measure to provide an alternative comparison perspective. A further issue may have been that T416 was only able to target for 10 of the 14 relevant documents; which could have affected precision, e.g. by large variances in document numbers being returned at different recall points which, based on low number of relevant documents, could have exaggerated precisions. Nevertheless, both Fig. 93 and Fig. 94 P&R measures show that the higher Orm mode *OQE* ratio of 4:21 produced the best precision results up to 50% recall level, in comparison to the Om (4:8) and Km modes.

4.2.2 Individual Query Set P&R Results

For the second part of the T416 review, the following P&R graphs demonstrate search effectiveness of individual queries, based on Ko vs. Oo vs. Oro and Km vs. Om vs. Orm query term search comparisons. The actual *OQE* terms for queries Q1, Q5 and Q10 are shown in Table 11, subsection 3.3.2. Both *optional* and *must-have* results *typical* of the group performances are shown next, followed by *atypical* results.

Typical Query Term Search Mode P&Rs

Fig. 95 and Fig. 96, based on query set Q1, and Fig. 97 and Fig. 98 based on Q8, demonstrate *typical* P&R profiles for each of the query search options, with higher Oro/Oo and Orm/Om precisions over Ko/Km found in 7 of 10 *optional* query sets and 5 of 10 *must-have* query sets.

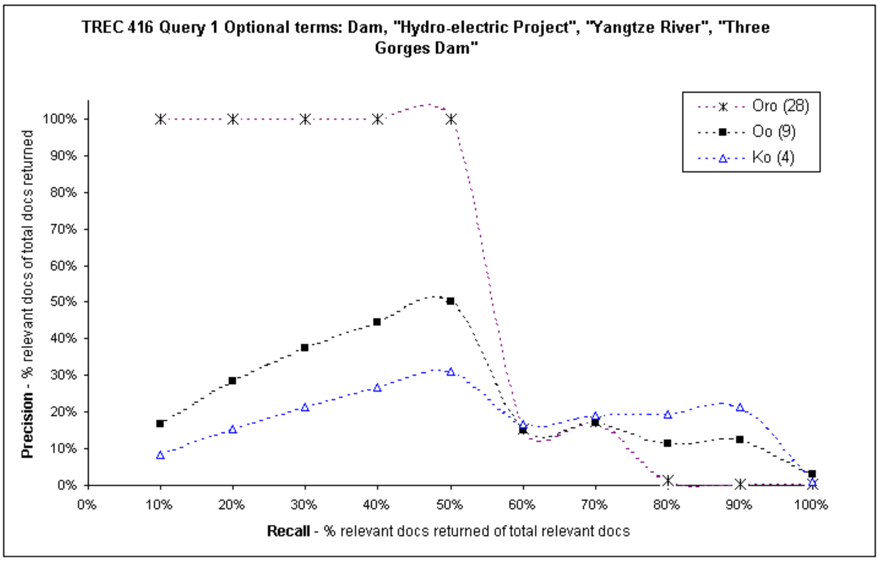


Fig. 95. T416 P&R for *optional* query Q1.

Fig. 95’s primary APV measures are 15% for Ko, 28% for Oo and 100% for Oro; showing good *S+S OQE* search effectiveness compared to keyword, whilst *S+S+R OQE* was excellent.

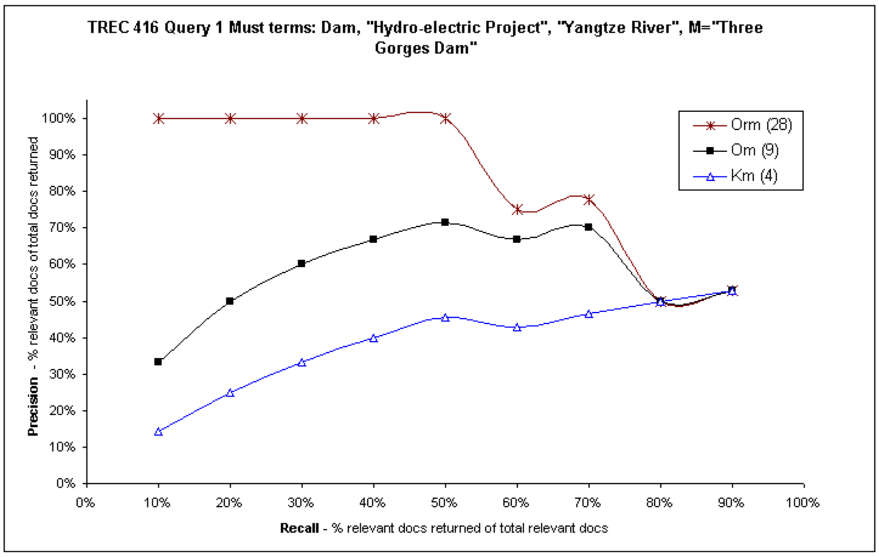


Fig. 96. T416 P&R for *must-have* query Q1.

Fig. 96’s *must-have* mode APV measures are 24% for Km, 48% for Om and 100% for Orm; again showing that *S+S OQE* search effectiveness was good and *S+S+R OQE* outcome was excellent, compared to keyword.

Fig. 97 and Fig. 98’s APV measures are very similar: Fig. 97 shows 19% for Ko, 38% for Oo and 64% for Oro, whilst Fig. 98’s *must-have* mode APVs are 22% for Km, 38% for Om and 64% for Orm. Both graphs showed the continued benefit of *OQE*, showing that *S+S OQE* search effectiveness was good and the *S+S+R OQE* outcome was very good, compared to keyword.

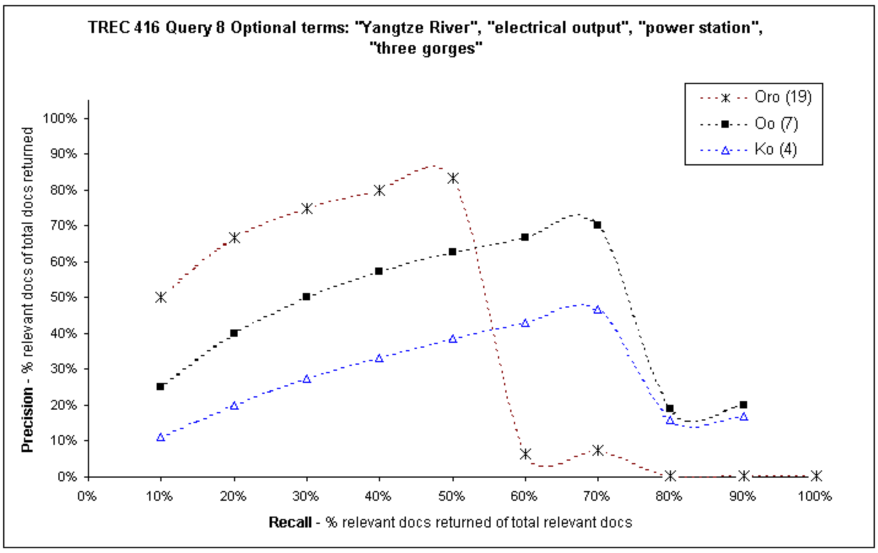


Fig. 97. T416 P&R for *optional* query Q8.

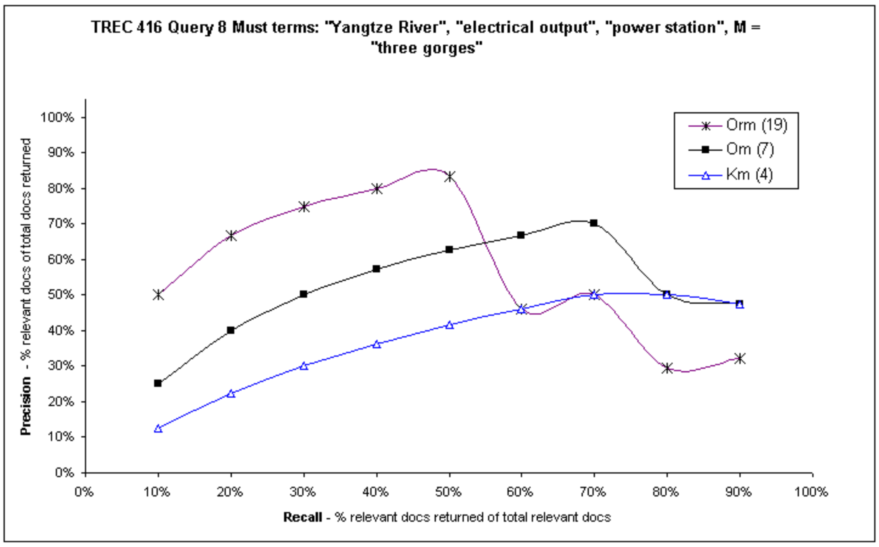


Fig. 98. T416 P&R for *must-have* query Q8.

Atypical Query Term Search Mode P&Rs

This subsection considers *atypical* query sets Q4, 5 and 10. First, virtually similar *atypical* profiles were evident in query sets Q4 (not shown) and Q5. Q5's results are shown in both Fig. 99 and Fig. 100; they show that Oro and Orm failed to produce higher precision than Oo/Om and Ko/Km respectively; in fact they badly underperformed.

Fig. 99 and Fig. 100's search effectiveness outcomes are very similar: Fig. 99's APV measures are 64% for Ko, 64% for Oo and 24% for Oro; whilst Fig. 100's *must-have* mode APVs are 100% for Km, 100% for Om and 48% for Orm. Both graphs show that *OQE* search effectiveness was at best no better (*S*+*S*), or poorer (*S*+*S*+*R*) compared to keyword.

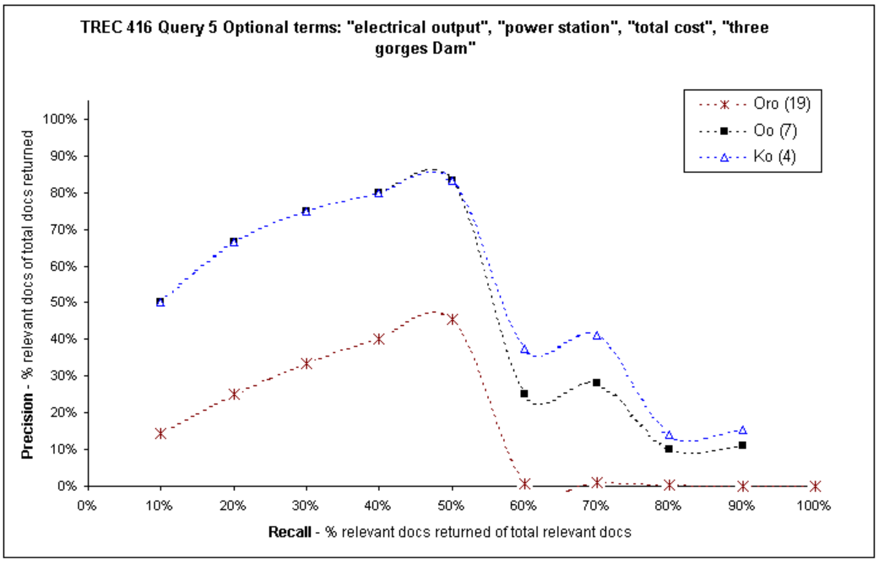


Fig. 99. T416 P&R for *optional* query Q5.

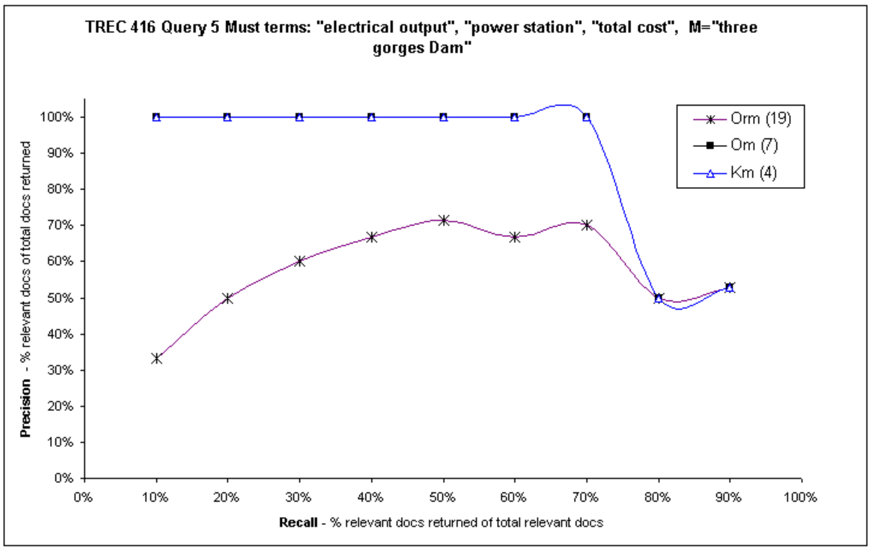


Fig. 100. T416 P&R for *must-have* query Q5.

A completely different result was found in the Q10 set, where the Ko, Oo and Oro P&R measures were the same (Fig. 101); this was repeated in Fig. 102's Km, Om and Orm modes.

Q10's Fig. 101 primary APVs are 7% each for Ko, Oo and Oro, whereas Fig. 102's *must-have* mode APVs are 48% each for Km, Om and Orm; these *OQE* outcomes were therefore unsuccessful compared to keyword only.

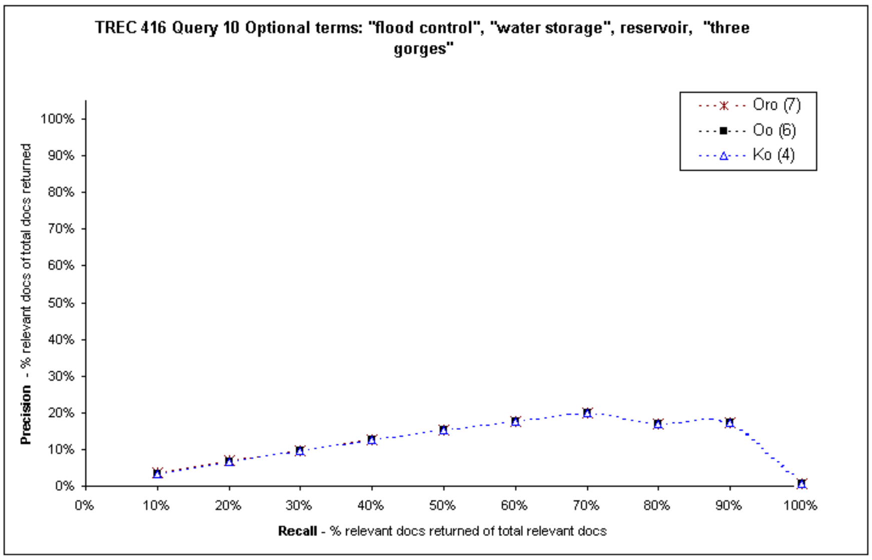


Fig. 101. T416 P&R for *optional* query Q10.

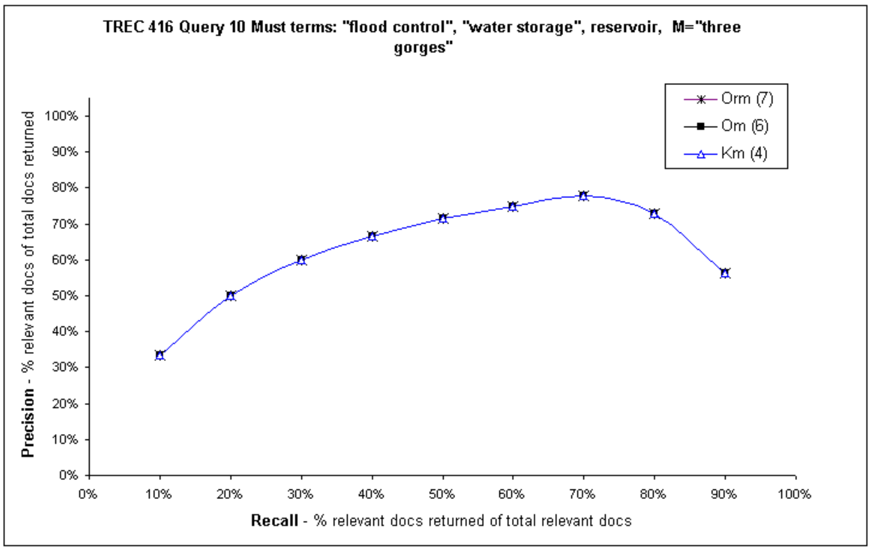


Fig. 102. T416 P&R for *must-have* query Q10.

A very low *OQE* ratio probably accounted for the Q10 result (4:6 for *S+S OQE* and 4:7 for *S+S+R OQE*), i.e. by minimising the impact of the *tf-idf* and concept weighting algorithms.

General Observations:

Individual query set APVs suggested a weak *S+S OQE* precision outcome; however, the grouped query results show *S+S+R OQE* convincingly outperformed both *S+S OQE* and keyword modes and both *OQE* modes produced the highest precision results up to 50% recall. Overall Ko and Oo recall was 95%, with Oro 100% (10 relevant documents in the query pool).

The Q1-10 query group *S+S OQE* generated on average 8 query terms, 7 of which were matched in documents, giving an *OQE* ratio of 4:8 and 88% class match. The *S+S+R OQE* generated on average 21 query terms, 20 of which were matched in documents, i.e. an *OQE* ratio of 4:21 and 95% class match. As in T401, both *OQE* ratios are high and suggest that the ontology context, again developed independently from the documents, proved very relevant for the query experiment. An analysis of *OQE* term matches is shown in Appendix E.

A further observation was the document *tf-idf* processing cost of *OQE*, where the extension of *S+S OQE* to *S+S+R OQE*, based on *optional* query terms (i.e. Oo and Oro), resulted in average returned documents per query increasing by 400% at each stage – see Table 18.

Table 18. T416 returned documents based on query mode.

Mode	Returned Docs.
Ko	2100
Oo	8500
Oro	33900

However, as the results have shown, P&R search effectiveness was not compromised.

4.2.3 Comparison of Precision Results Across All Query Modes

Fig. 103 shows the search effectiveness comparisons plotted in radar graph for 10%, 20% and 30% recall points, this time comparing the six query modes. The scoring was thus based on 1-6 points, e.g. if Q1 resulted in the following precisions at 10% recall: Ko 30%, Oo 50%, Oro 70%, Km 60%, Om 40% and Orm 50%; the query term search modes would be awarded 1, 4, 6, 5, 2 and 4 points respectively.

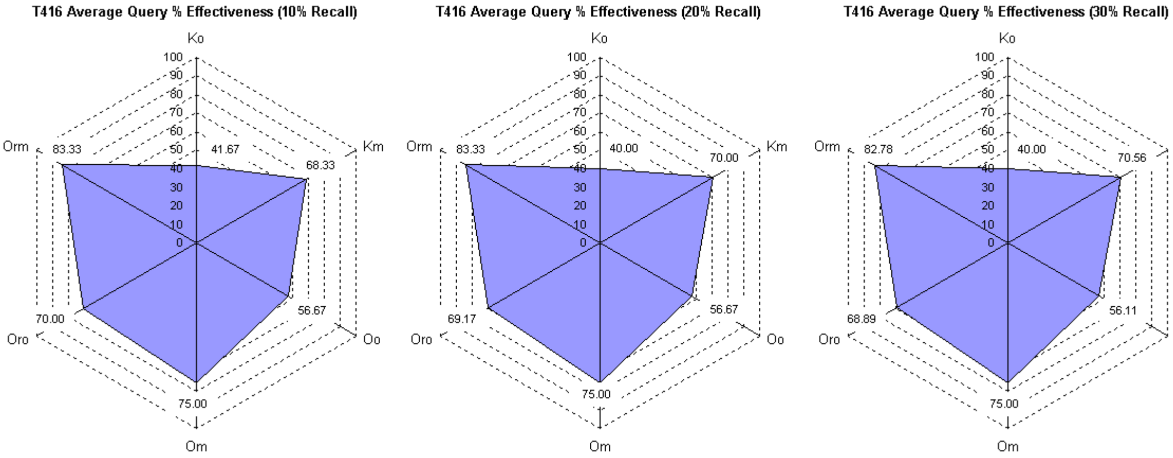


Fig. 103. T416 average query percentage effectiveness.

The performance of all modes show no real change in search precision between the 10% to 30% recall points; however, the weakest performance was keyword Ko mode and the strongest were Om, Oro and Orm *OQE* modes. *Must-have* modes (Km, Om and Orm) performed better than their *optional* mode counterparts – probably reflecting the distinct query topic focus on the Three Gorges theme. Highest precisions tended to be Om, Oro and Orm modes; indicating that *S+S+R OQE*, with relation classes specified in asserted conditions, can optimise results.

4.2.4 APV Measures

The APV measures for specific individual query set outcomes have been presented in subsection 4.2.2. The APVs for the 20 individual queries, i.e. 10 query sets in both *optional* (Ko/Oo/Oro) and *must-have* (Km/Om/Orm) modes, are shown in Appendix I. Table 19 summarises the times a mode's APV was most successful (Top) or performed the same (Tied).

Table 19. Comparisons of T416 query mode APV successes.

Optional Mode				
% Ko Top	% Oo Top	% Oro Top		% Tied
0%	10%	60%		30%

Must-have Mode				
% Km Top	% Om Top	% Orm Top		% Tied
10%	0%	50%		40%

The results show that *OQE* was most search effective in 70% of *optional* queries and 50% of *must-have* queries; with Oro and Orm predominant. Whilst 30% of *optional* queries and 40% of *must-have* queries showed tied outcomes, *OQE* performance was good overall and demonstrated the benefit of *S+S+R OQE*.

4.2.5 Comparing *Optional* and *Must-have* Query Mode Successes

In comparison to Table 19 above, Table 20 shows the number of times each *optional* and *must-have* search mode produced the highest precision at 10, 20 and 30% recall points in the 10 query sets. Tied precision outcomes are shown separately.

In T416 there was a high incidence of tied query mode results, i.e. 23% in *optional* and 40% in *must-have*. The tied outcomes do not necessarily mean that all three modes were equal. In *optional* mode the 2 tied results were between Ko and Oo; in *must-have* mode, 3 tied results were between Km and Om, and the other was tied by all three modes.

Table 20. Comparisons of T416 query mode successes.

Optional Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Ko	0	0	0	0.0	0%
Oo	1	1	1	1.0	10%
Oro	7	7	6	6.7	67%
Tied Results	2	2	3	2.3	23%

Must-have Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Km	1	1	1	1.0	10%
Om	0	0	0	0.0	0%
Orm	5	5	5	5.0	50%
Tied Results	4	4	4	4.0	40%

Whilst the *OQE*-based precision performance was less dominant than in the T401 results, Oro and Orm produced the best average results over their Ko/Oo and Km/Om comparators (67% and 50% respectively). The 30% recall point was equivalent to measuring performance in returning the highest scoring 3 of 10 relevant documents and Oro mode performed best of all. In terms of overall results, *must-have* mode was relatively less favourable for *OQE* than *optional* mode, i.e. Om with Orm at 50% (vs. Km 10%), compared to Oo with Oro at 77% (vs. Ko 0%).

Overall, the results agree with the APV outcomes in subsection 4.2.4, with good *S+S+R OQE*.

These results are illustrated in Fig. 104 (tied are Ko~Oo~Oro or Km~Om~Orm).

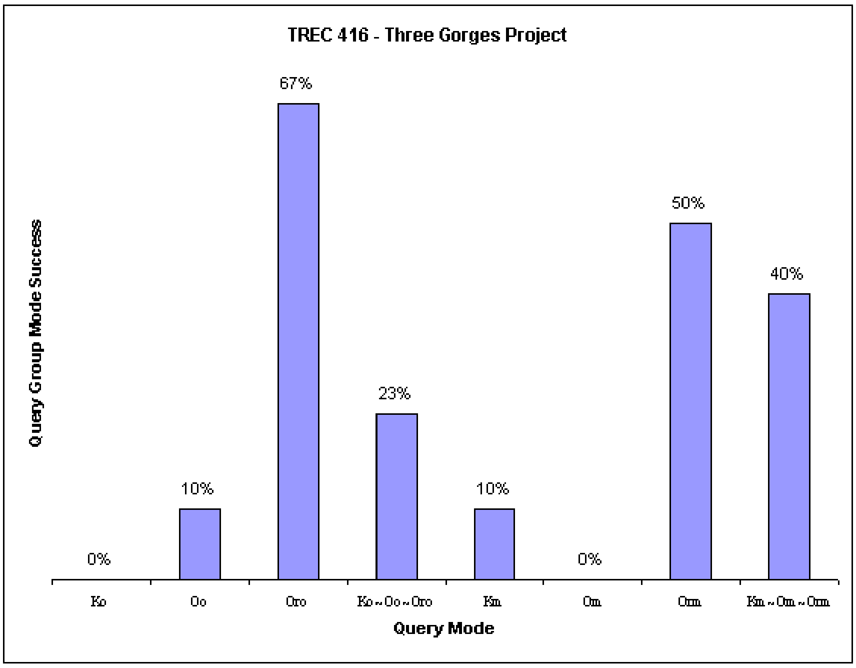


Fig. 104. T416 query mode successes.

4.2.6 Critical Review of Experiment

As the T416 ontology supported 58 query terms, with a similarly shallow hierarchy to T401, the incremental *relation* class expansion provided a useful comparator for evaluating *OQE*; see subsections 3.2.3 (algorithm), 3.2.5 (relation weights) and 3.3.2 (Fig. 66).

The experiment was successful in demonstrating *OQE*-based search effectiveness improvement: the APV summary (subsection 4.2.4) shows improved APVs in 70% of *optional* mode *OQEs* and in 50% of *must-have* mode *OQEs*; primarily with Oro and Orm. Whilst 30% of *optional* queries and 40% of *must-have* queries had tied outcomes, *OQE* performance was good overall. The *S+S+R OQE* outcomes provided some further learning: that a wider, relation class expansion can provide strong precision improvement over *S+S OQE*.

As highlighted, the greatest incidence of joint query successes was evidenced by keyword and *S+S OQE* modes (i.e. Ko with Oo and Km with Om); this would initially suggest that, whichever of the modes are selected, the other might need to be considered as a secondary search option. Further learning was that *S+S+R OQE* could have a considerable processing overhead for a search tool, given the incremental 400% increases in document hits highlighted in Table 18.

Optional mode *OQE* generated consistently high recall outcomes: on average 95 % of the 10 relevant documents were found using Ko and Oo modes, whilst Oro mode achieved 100% recall; this suggested that an Oro mode approach was more "recall effective" than both Ko and Oo. As highlighted in T401, *must-have* terms can be problematic as potentially relevant documents could be excluded by, in effect, using an explicit restriction; this was demonstrated in T416, where the three *must-have* modes (Km, Om and Orm) only achieved 81% recall.

Therefore, unless a *must-have* term correlates precisely with the query objective, it would appear that *optional* mode queries are more likely to provide better APV outcomes; by allowing the *tf-idf* and *OQE* weighting algorithms to have an unrestricted effect, i.e. fully harvest all potential documents.

The density of the 10 relevant documents in the document set WT01-WT19 was approximately 0.006% (10/160838), compared to an average 0.05% for the overall document set for the 3 experiments (WT01-WT19); this represented a very small target set, although only 14 documents in total were relevant for the T416 query topic. Whether or not the small number of relevant documents could have had an effect on the overall precision scores is not clear, as precision would also be dependent on the frequency and spread of the much larger non-relevant document set. However, as mentioned earlier, the low number of relevant documents in the query pool can have an exaggerated effect on average precision based on returned document numbers, e.g. Fig. 92, where a query returned disproportionately higher numbers at certain recall points.

The low average keyword to *S+S OQE* ratio of 4:8 provided less opportunity for the *tf-idf* and weighting algorithm to have an effect on relevance scores; the use of flat ontology hierarchy made this more likely. When *S+S+R OQE* (Oro and Orm) was used, the asserted condition relation classes produced markedly better APV outcomes, to counteract the flat hierarchy.

Finally, as mentioned in the T401 review, the *OQE* results may have been improved if different weightings had been applied; as indicated by the later T401 *S+S* weight reversal experiments in subsection 4.4.3.

4.2.7 Reflections on Hypotheses

Comments are now provided for the hypotheses proposed in subsection 1.9.2.

Hypothesis (i) – “hierarchical (*S+S*) *OQE* can have a positive impact on precision and recall, although class hierarchy expansions alone may not produce optimal results. Query term-matched classes may have more beneficial wider semantic *relations* with other classes” (*S+S+R OQE*).

The T416 experiment is interesting: *OQE* has achieved the highest APVs in 60% of all query comparisons, predominantly in Oro and Orm query modes; also Oo was higher than Ko but Km was better than Om. Nevertheless, the key issue is that the APV outcomes have successfully demonstrated the benefits of *OQE* and the improved search effectiveness achieved using *S+S+R OQE*. A secondary benefit is that, in the 10% to 50% recall range, Oro and Orm precisions were approximately 10 percentage points above Oo and 20 points above Om, respectively; these outcomes have clearly demonstrated the benefits of extending the query expansion beyond simply sub and super classes.

Overall recall was slightly better in *optional* mode, where *S+S+R OQE* had a good impact, with Oro achieving 100% recall compared to Ko and Oo (95%). In *must-have* mode, recall was 81% for all modes, probably reflecting that poorly chosen *must-have* terms can be counter productive.

The *OQE* APV results have been good, particularly *S+S+R OQE*, and provide clear evidence to support the hypothesis, in terms of both precision and recall.

Hypothesis (ii) - "higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes".

As mentioned in T401, this will be evaluated by comparing the initial T401 weightings and APV outcomes with revised T401 weighting experiment outcomes in subsections 4.4.1 to 4.4.3.

Hypothesis (iii) – "topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion, despite their potentially restrictive coverage, … as opposed to contextually wider, or more comprehensive, ontologies …. to avoid superfluous query expansion".

The Hydro-electric ontology context was developed specifically for the T416 topic and the average *OQE* term class matching was 88% for Oo and 95% for Oro (the highest of the three query experiments). The context provided a good platform for effective *OQE*, given the overall positive effect of improving APV outcomes; these results, combined with improved precision up to 50% recall, provide good evidence to support the hypothesis.

The merit of using a topic specific ontology context, to maximise contextual relevance and minimise "superfluous" *OQE*, will be further evaluated in subsection 4.4.4: by comparing the T416 APV and class matching results against T401 and T438.

4.3 T438 'TOURISM, INCREASE' EXPERIMENT RESULTS

The following T438 P&R graph reviews were based on 4 query modes to provide 2 *two-way* comparisons in each of 20 queries, i.e. Ko vs. Oo *optional* and Km vs. Om *must-have* search options, as in T401. The document set cut-off at WT12 provided for 36 of 46 pool relevant documents to be targeted across 96,885 documents.

The query matrix shown in Table 12, subsection 3.3.3, was based on all of the 20 query sets comparing keyword against *S+S OQE*. The results are considered, firstly by examining the overall *group* of query sets and then by considering *typical* and *atypical* P&R results of *individual* query sets. It should be noted that, unlike in T401 and T438, because of the mix of results, *typical* will be used to describe where any distinct groups of individual graphs show commonalities, as opposed to representing the mode of the group of queries.

As previously highlighted, the T438 graph y-axis (precision) scales have been reduced for presentation purposes, i.e. a maximum 10% scale y-axis is used for grouped P&R results in subsection 4.3.1 and a maximum 30% scale is used for individual query set P&Rs in subsection 4.3.2. The scales are dependent on the maximum precision values returned, e.g. many are below 10%.

4.3.1 Overall Group Query Term Search Mode P&Rs

This subsection reviews search effectiveness based on the combined results of the two Q1-20 query groups of *optional* and *must-have* queries. Given the very low precision levels, the y-axes display a 0% to 10% precision scale, for better visual understanding.

Optional Query Modes

The P&R graph for query group Q1-20 *optional* Ko and Oo modes are shown in Fig. 105; the MEA-based comparison measure is provided in Fig. 106.

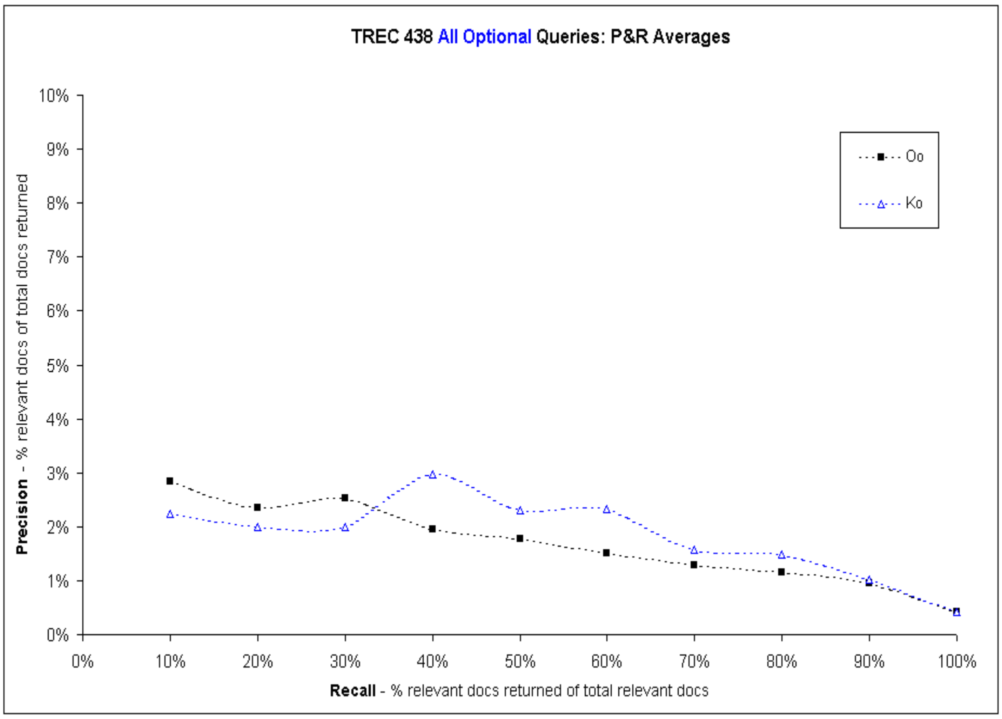


Fig. 105. T438 overall P&R for *optional* queries.

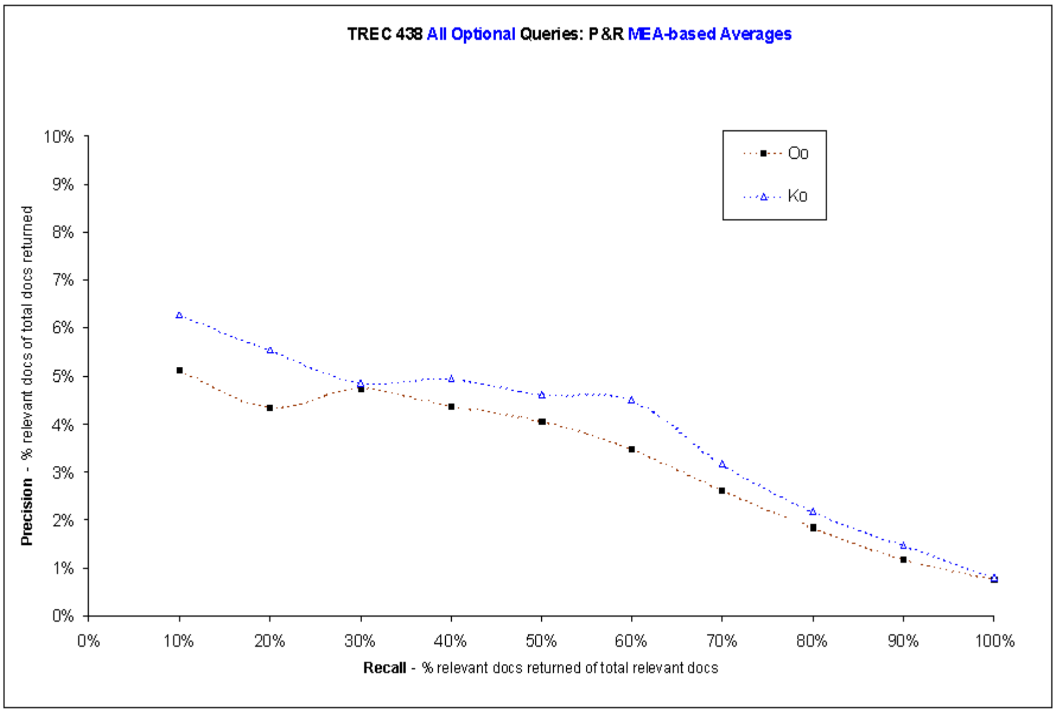


Fig. 106. T438 overall P&R for *optional* queries - MEA measure.

The precision results in Fig. 105 show Oo search option produced a slightly better average performance up to 30% recall, although the APV was only 2.1% for Ko and 2.6% for Oo. In contrast, Fig. 106 shows that the MEA-based APVs were 5.6% for Ko and 4.7% for Oo.

The two graphs present conflicting Oo APV results over Ko; they were either negligible or adverse. *OQE* search effectiveness should be seen as inconclusive.

Must-have Query Modes

The combined Q1-20 results for the *must-have* Km and Om search options are shown in Fig. 107 and the MEA measure appears in Fig. 108. The P&R profiles show very little difference overall between Km and Om and low performance in both P&R comparison measures.

Primary outcomes are: Fig. 107's APVs show negligible search effectiveness with 1.7% for Km and Om; MEA-based Fig. 108 provides a similar outcome, but with 4.9% for Km and 4.3% for Om. Results are inconclusive, although Om was higher (6%) at 10% recall.

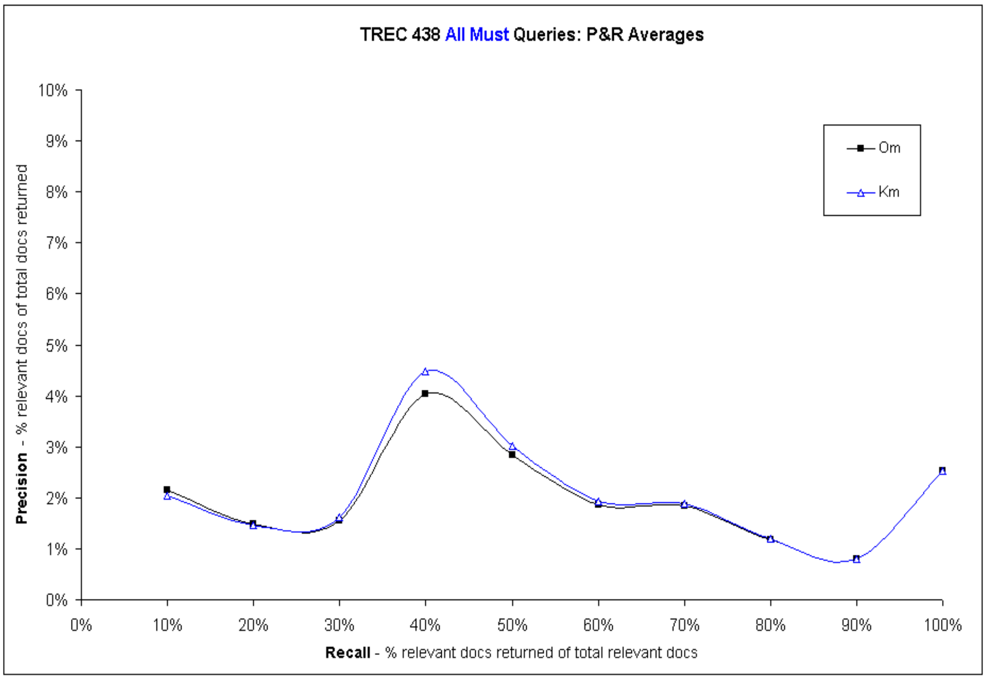


Fig. 107. T438 overall P&R for *must-have* queries.

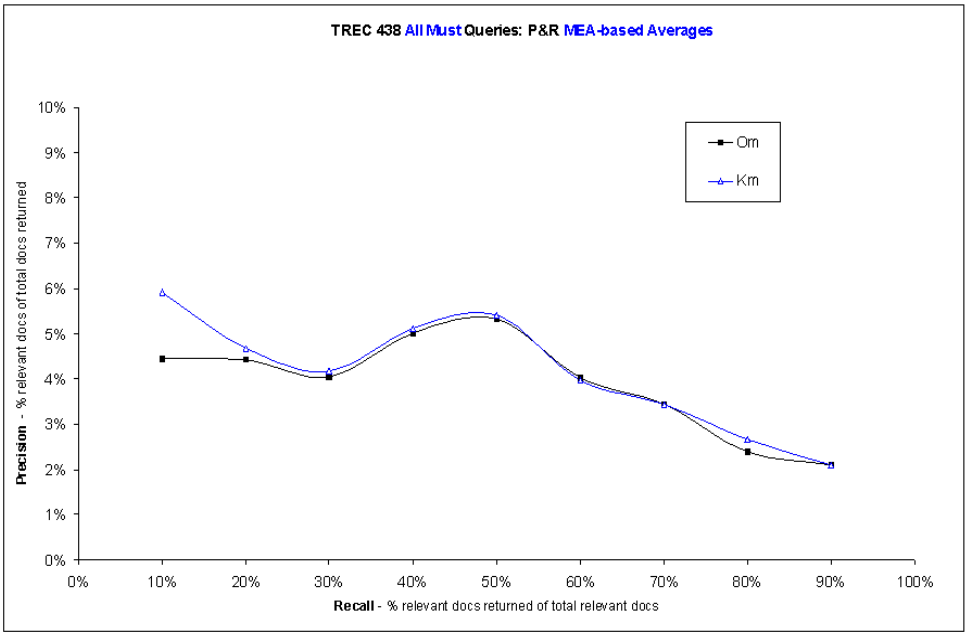


Fig. 108. T438 overall P&R for *must-have* queries - MEA measure.

4.3.2 Individual Query Set P&Rs

The following P&R graphs demonstrate individual query set search effectiveness based on Ko vs. Oo and Km vs. Om comparisons. The graphs use a 0% to 30% precision y-axis, for better visual understanding. The graphs convey an inconsistent message because one mode may be less effective at say 10% recall than at 30% recall. APV measures helped to address this issue.

Typical Query Mode P&Rs

For *optional* queries, *typical* results were somewhat polarised between keyword and *OQE* based modes. Query sets Q5, 6, 10 and 12 were generally characterised by higher Oo precision and much greater recall over Ko, as demonstrated in the Q5 Fig. 109 and Q12 Fig. 110 graphs.

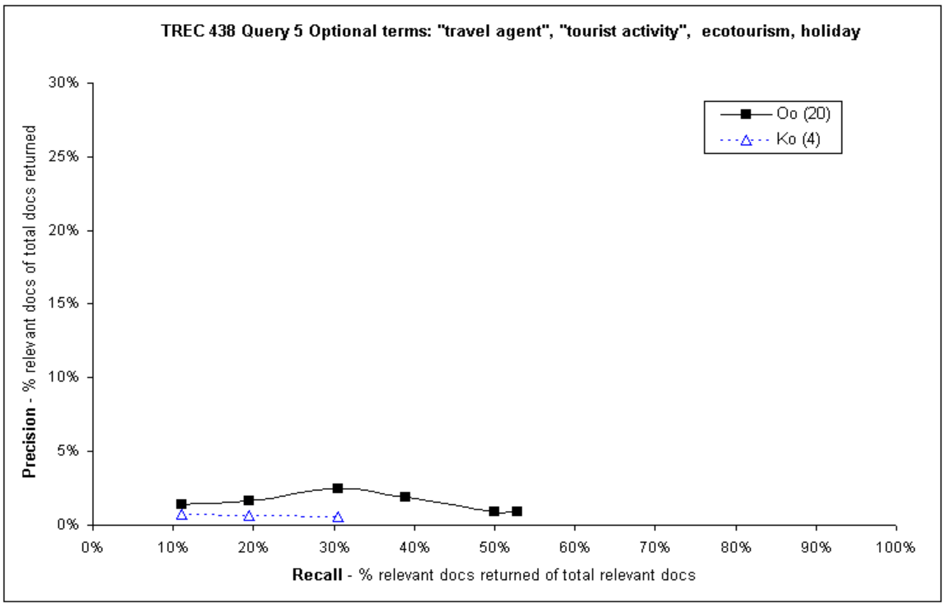


Fig. 109. T438 P&R for *optional* query Q5.

Fig. 109 primary APV measures are 0.7% for Ko and 1.8% for Oo. Despite the low performance, *OQE* search effectiveness provided a modest improvement. Similar results were achieved for Fig. 110 APVs: 0.5% for Ko and 2.7% for Oo. Both *OQE*s extended recall.

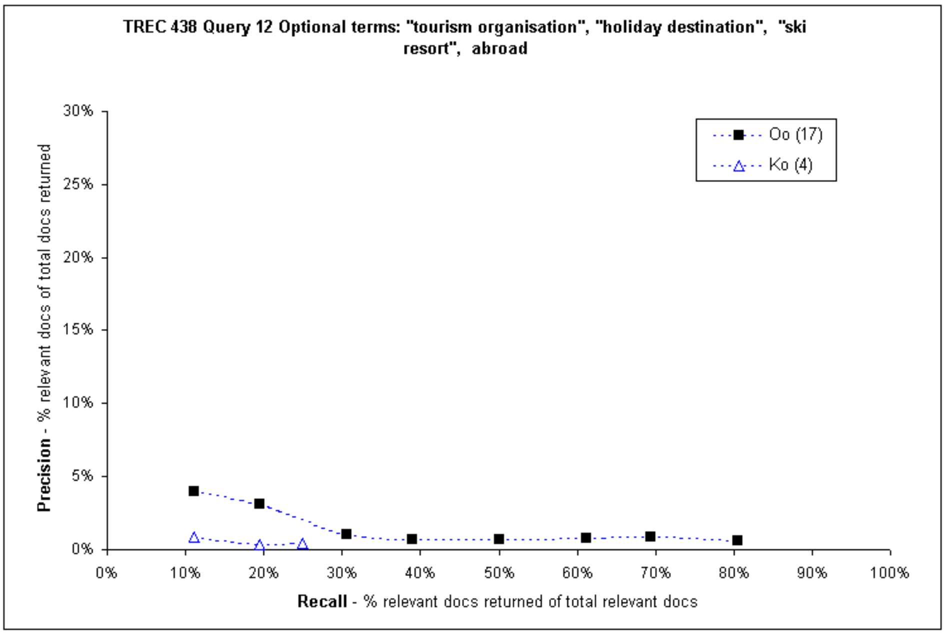


Fig. 110. T438 P&R for *optional* query Q12.

Conversely, the *optional* query sets for Qs 15, 17, 19 and 20 were generally characterised by higher Ko precision and recall over Oo; as shown by Q15 (Fig. 111) and Q19 (Fig. 112).

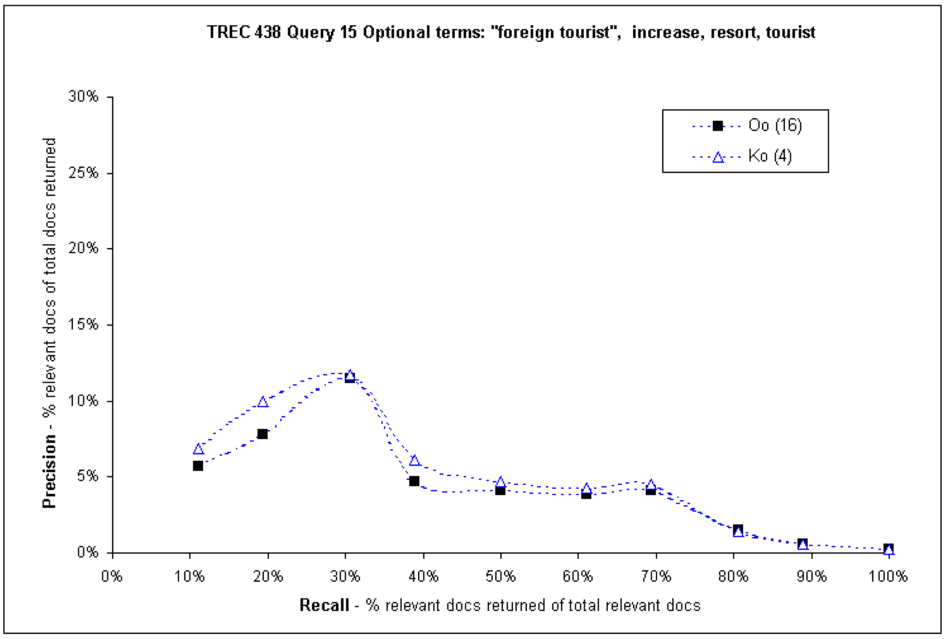


Fig. 111. T438 P&R for *optional* query Q15.

Fig. 111 APVs are 9.5% for Ko and 8.3% for Oo, whilst Fig. 112's are 12% for Ko and 9% for Oo. Both graphs exhibit poor *OQE* search effectiveness and bad Oo performance versus Ko.

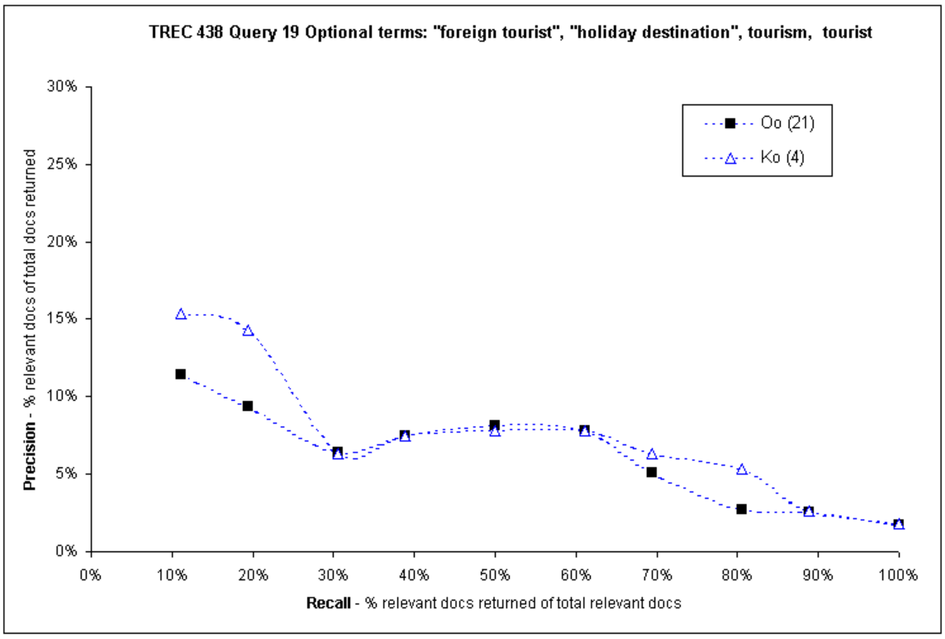


Fig. 112. T438 P&R for *optional* query Q19.

The *optional* graphs, for Qs 3, 7, 14, and 16, each produced very similar Ko and Oo precision and recall outcomes, as shown below in Q3, Fig. 113; whilst queries Q2, Q4, Q11, Q13 and Q18 demonstrated variable precision performance but similar recall, as in Q4, Fig. 114.

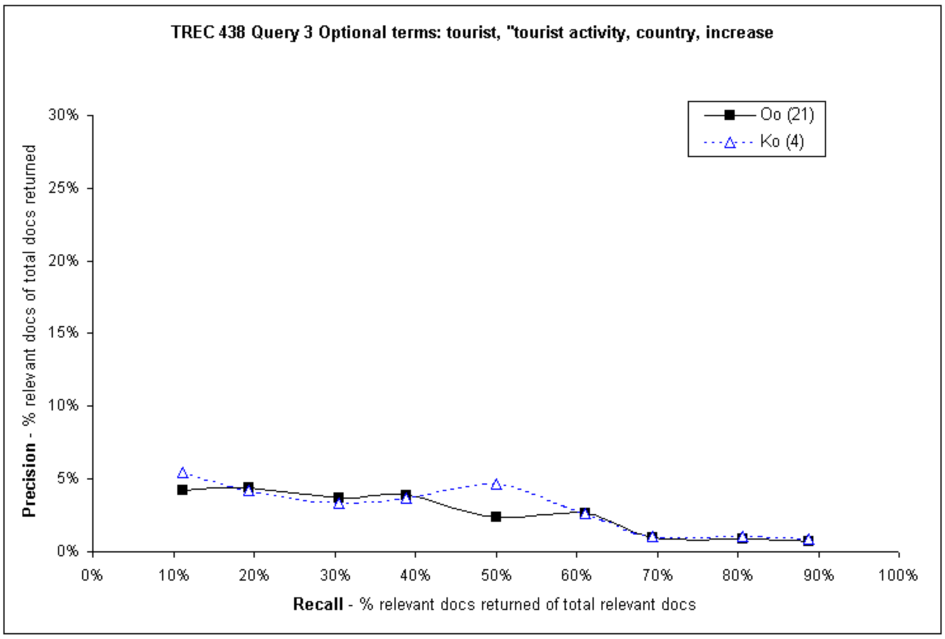


Fig. 113. T438 P&R for *optional* query Q3.

Fig. 113 APVs are 4.4% for Ko and 4.1% for Oo; this showed *OQE* was ineffective. Fig. 114 outcomes were 7.6% for Ko and 9.2% for Oo, providing only modest *OQE* improvement.

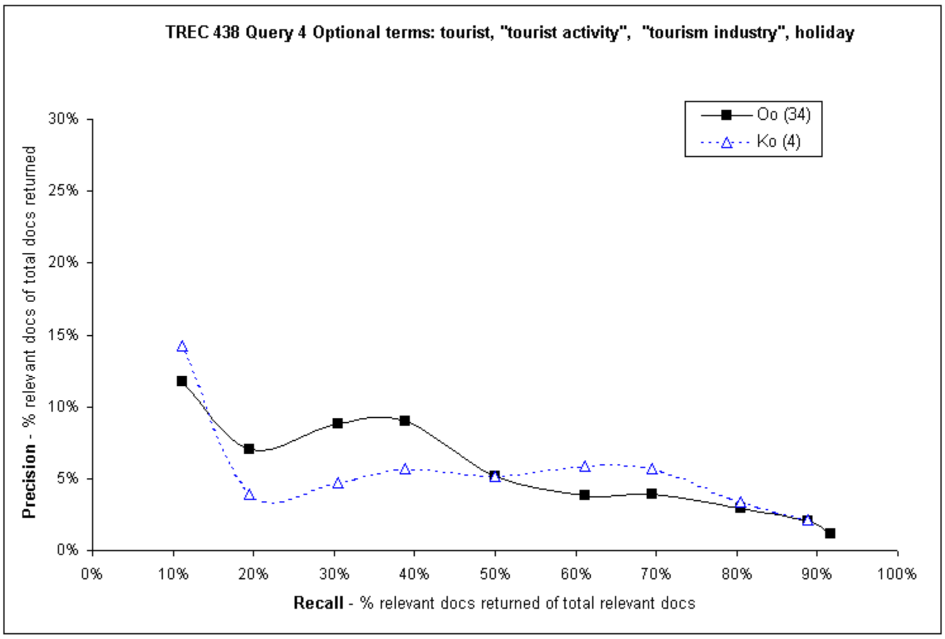


Fig. 114. T438 P&R for *optional* query Q4.

For *must-have* queries, the majority (Qs 3, 7, 11, 14-16, 18, 20) produced similar patterns of varying precisions but with similar high levels of recall, as in Q11, Fig. 115. The remainder, Qs 1, 4-5, 8-9, and 12-13 gave very low recall levels of generally less than 30%, as in Q5, Fig. 116.

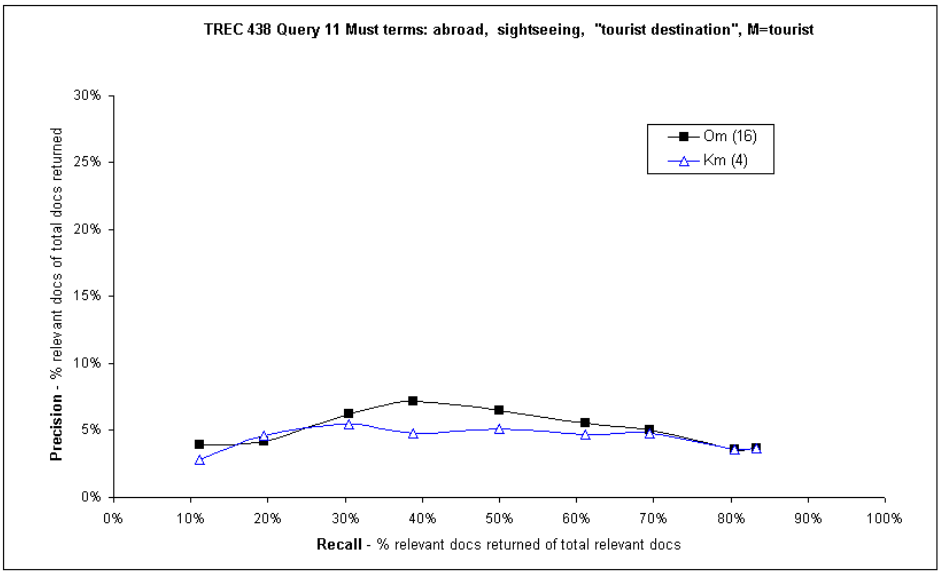


Fig. 115. T438 P&R for *must-have* query Q11.

Fig. 115 shows 4.3% for Km and 4.8% for Om. Fig. 116 shows 0.6% for Km and 1.1% for Om. The APVs indicate that *OQE* search effectiveness improvement was negligible.

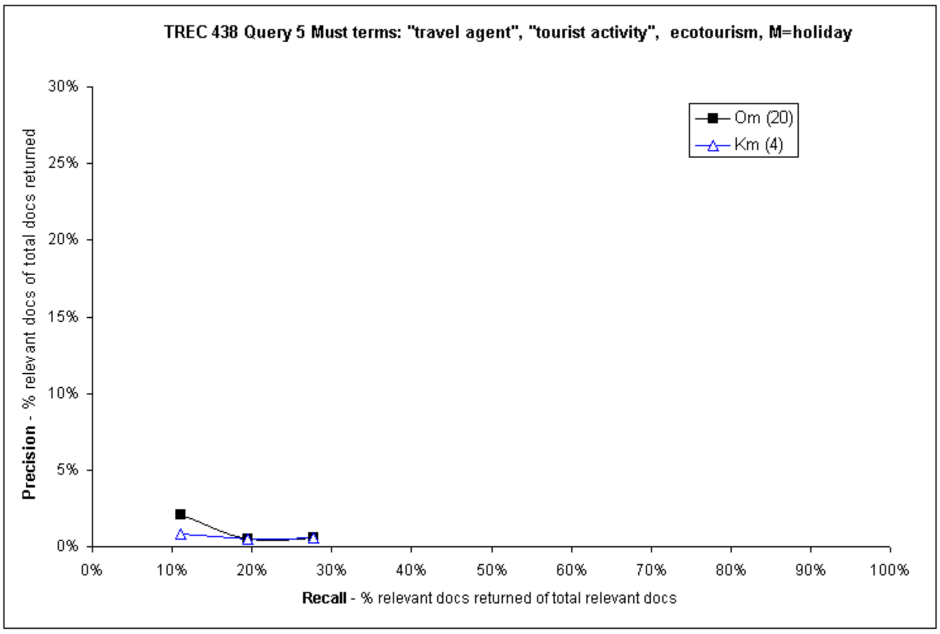


Fig. 116. T438 P&R for *must-have* query Q5.

Atypical Query Mode P&Rs

For *optional* queries, *atypical* results were found in Q1, 8 and 9, where higher precision was noted in the Ko option whilst higher recall was noted in Oo, see Q1, Fig. 117 and Q8, Fig. 118.

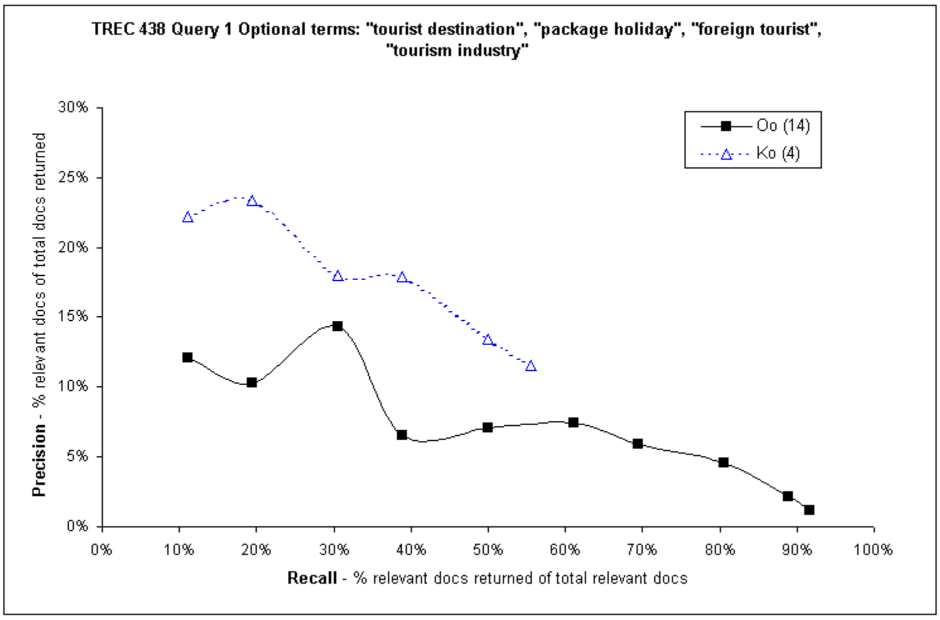


Fig. 117. T438 P&R for *optional* query Q1.

Fig. 117 primary APVs are 21% for Ko and 12% for Oo; Q8's Fig. 118 APVs are 9.9% for Ko and 6.2% for Oo. Both *OQE* queries show a bad APV outcome and adverse search effectiveness.

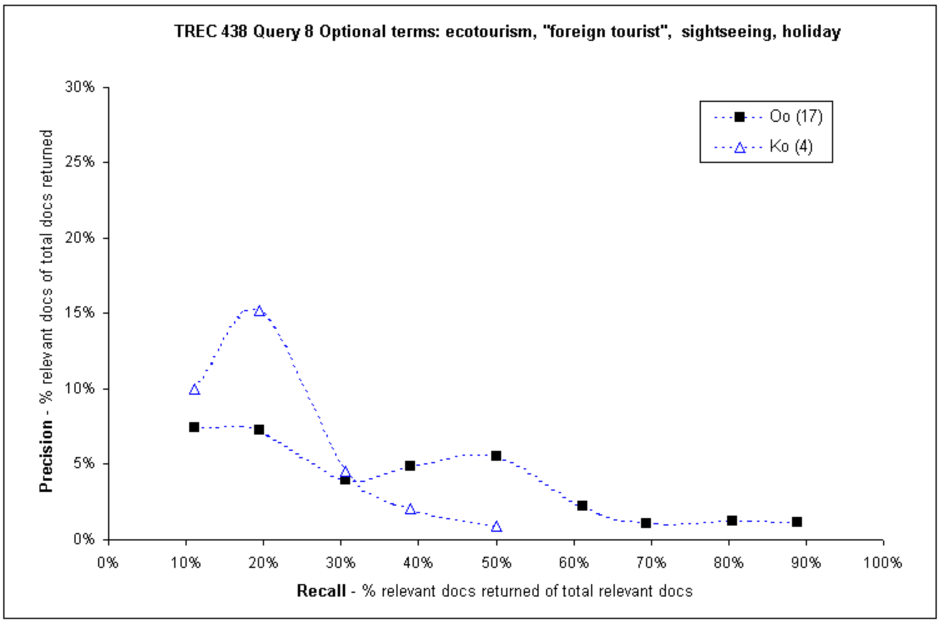


Fig. 118. T438 P&R for *optional* query Q8.

For *must-have* queries, Q2, Q6 and Q10 returned no documents, clearly reflecting an inappropriate *must-have* term, but Q17 and Q19 had a better Km precision, as in Q17, Fig. 119. Q17's APVs are 6.7% for Km and 4.2% for Om. Similar to Fig. 117 and Fig. 118, the *OQE* query shows a poor APV outcome and adverse search effectiveness.

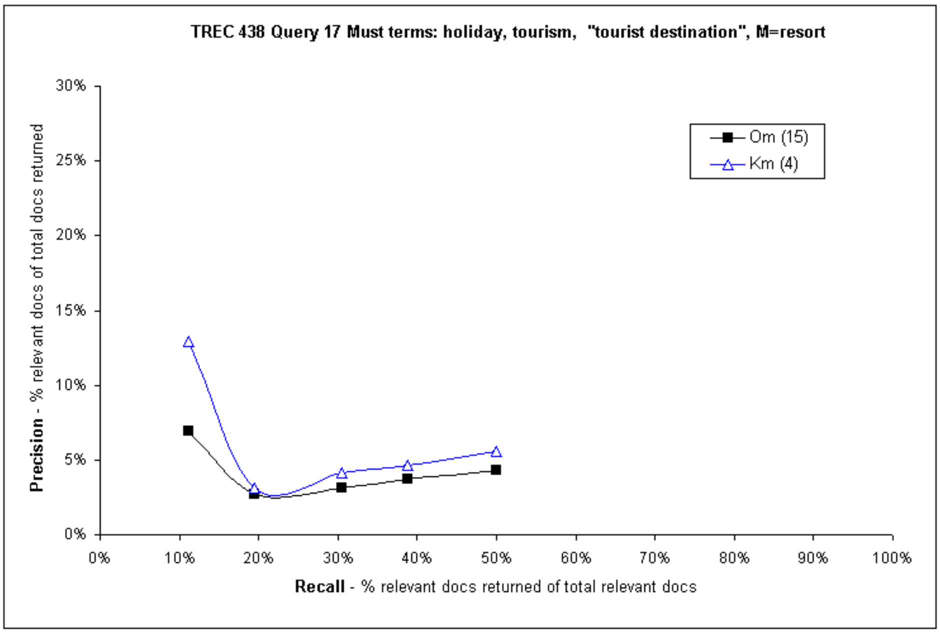


Fig. 119. T438 P&R for *must-have* query Q17.

General Observations:

The *S+S OQE* for the query group Q1-20 generated on average 16 query terms per query, i.e. an *OQE* ratio of 4:16, of which 13 (81%) were matched in the documents. Whilst the T438 multi-context ontology had classes with direct Tourism relevance, the actual Tourism context element was not developed specifically for the T438 topic experiment; this may have been a reason why the *OQE* term matching ratio was noticeably lower than T401 (89% for Oo/Om) and T416 (88% for Oo/Om and 95% for Oro/Orm) - see analysis of *OQE* term matches in Appendix E.

It was initially thought that the lower rate might have been attributed to T438's ontology being a wider (multi-context) ontology but an examination of unmatched terms found that only a small number were related to the other contexts imported into the ontology. Further, there was only a small number of subclass to super class mappings between Tourism classes and other context classes; consequently, the multi-context characteristic was not considered relevant.

T438's precision rates were consistently lower than those achieved in the previous experiments: the highest was approximately 6%, whereas T401 and T416 were between 50% and 80%. On average, 74% recall (of the 36 relevant documents) was achieved using Ko mode but Oo mode was 90%, whereas only 53% recall was achieved in Km and Om modes; whilst a much higher Oo recall over Ko was demonstrated, the experiment produced the lowest recall rate of the three query topics. The results serve to demonstrate both the exclusion effect (using an imprecise *must-have* term) and that *optional* mode queries can produce better P&R outcomes (by allowing the *tf-idf* and *OQE* weighting algorithms to have an unrestricted effect).

The density of the 36 relevant documents in the document set (WT01-WT12) was approximately 0.037% (36/96885) compared to 0.05% for the overall document set (WT01-WT19); this was the highest of the three experiments.

4.3.3 Comparison of Precision Results Across All Query Modes

Search effectiveness comparison results are shown in the radar graphs in Fig. 120 and are based on the four query modes being awarded 1-4 points for recall points 10-30%. This was repeated for all query sets and the average score for each mode calculated as a percentage of maximum available 4 points.

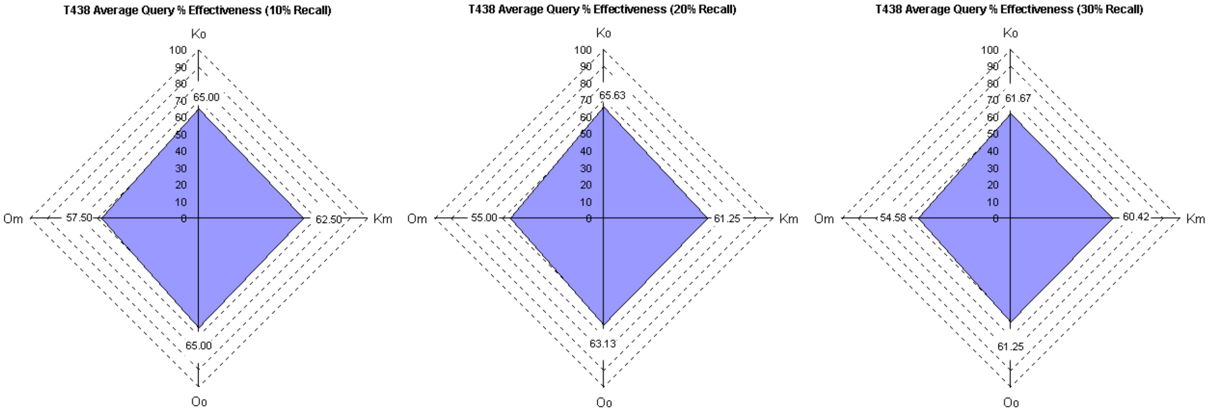


Fig. 120. T438 average query percentage effectiveness.

Again, the performance profiles show a gradually reducing search precision performance that is consistent across all modes between the 10% to 30% recall points. In this experiment, the three graphs show marginally better results were achieved in *optional* query term searches (Ko and Oo), followed by *must-have* (Km), with Om performing least effectively. However, the experiment results fail to markedly differentiate between the query term search options and have therefore been inconclusive; this may be a reflection on the generic nature of the query topic relevance guidelines, thereby making it more difficult to contextually apply constraints to specific query terms, e.g. when using the *must-have* searches.

4.3.4 APV Measures

Specific individual query set APV outcomes have been presented in subsection 4.3.2. Appendix I provides APVs for all 40 queries, i.e. 20 query sets in both *optional* (Ko/Oo) and *must-have* (Km/Om) modes. Table 21 summarises when a mode's APV was either most successful (Top) or performed the same (Tied).

The results show that *OQE* was most search effective in only 30% of *optional* queries and 35% of *must-have* queries; keyword mode was predominant in 65% and 55% of queries respectively and tied outcomes were evident in 5%/10% of queries.

Table 21. Comparisons of T438 query mode APV successes.

Optional Mode			
% Ko Top	% Oo Top		% Tied
65%	30%		5%

Must-have Mode			
% Km Top	% Om Top		% Tied
55%	35%		10%

Overall, the *OQE* results were disappointing, with some adverse precision outcomes and poor search effectiveness demonstrated; although *OQE* was a better solution in a third of queries.

4.3.5 Comparing *Optional* and *Must-have* Query Mode Successes

In comparison to Table 21 above, Table 22 shows the number of times each *optional* and *must-have* search mode produced the highest precision at 10, 20 and 30% recall points in the 20 query sets. Tied precision outcomes are shown separately.

The results show that *OQE* precision performance was generally weaker than the keyword-only: at the 30% recall point, i.e. equivalent to returning the highest scoring 10 of 36 relevant documents, the Ko and Km results were each 50%, with Oo at 40% and Om at 30%. On average the Ko and Km modes produced the best overall results (53% and 50% respectively) over Oo (43%) and Om (32%). The average incidence of modes achieving joint highest (tied) scores was highest in *must-have* mode, with Km/Om at 18% compared to Ko/Oo at 4%.

Table 22. Comparisons of T438 query mode successes.

Optional Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Ko	10	12	10	10.6	53%
Oo	10	8	8	8.6	43%
Tied Results	0	0	2	0.7	4%

Must-have Mode	10% Recall	20% Recall	30% Recall	Average	Query group average success rate
Km	11	9	10	10.0	50%
Om	8	5	6	6.3	32%
Tied Results	1	6	4	3.7	18%

The results show that *OQE* was the least effective of the 3 query experiments; they support the poor *OQE* APV outcomes shown in subsection 4.3.4. However, whilst Table 21 and Table 22 show keyword search as more effective, the number of Oo and Om successes cannot be ignored. Based on the initial relevance weightings applied, T438 outcomes suggest the need for both keyword and *OQE* search processes in a search engine interface, i.e. a two-stage search process could provide incrementally beneficial results.

These results are illustrated in Fig. 121 (tied are shown as Ko~Oo or Km~Om).

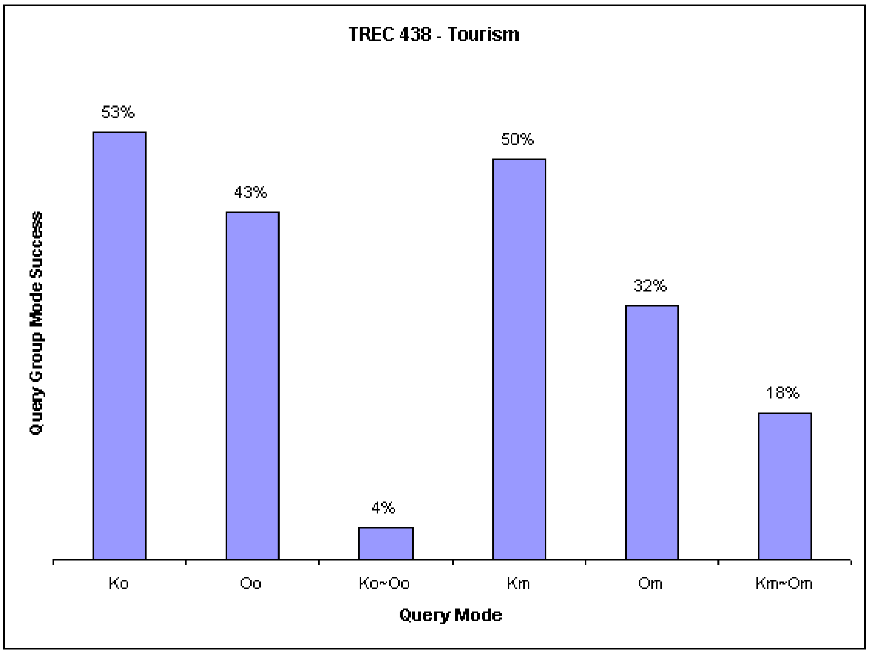


Fig. 121. T438 query mode successes.

4.3.6 Critical Review of Experiment

Overall, the experiment failed to demonstrate a consistent and satisfactory *OQE* search effectiveness improvement: the APV summary (subsection 4.3.4) shows that, on average, only a third of all queries were improved by *OQE*, with keyword mode predominant in approximately 60% of queries. The *OQE* APV results demonstrated poor search effectiveness on balance; failing to achieve the APV levels and successes identified in T401 & T416.

The experiment's low precision and recall outcomes (compared T401 and T416) might possibly suggest poor query term selection but this should have applied across all modes; and yet, Oo mode achieved a satisfactory 90% recall.

Some critical success factors have been identified for consideration and relate to three issues: that the ontology was not designed for purpose; the experiment used only the base *S+S OQE*; and the topic statement had a vague query objective.

The multi-context ontology was not developed specifically for the T438 topic and some of the class names could be considered poorly formed, i.e. using complex names, e.g. holiday classes: "Sun Sea and Sand Holiday", "Health Spa and Relaxation" and "Fly-Cruise Holiday"; tourist activities: "Exploring Locations" and "Booking Accommodation"; tourism organisations:

"Regional Tourist Board" and "Tourism Skills Training and Development Organisation". A valid criticism would be that they should have been constructed, using modular (*primitive*) concepts to describe complex (*defined*) classes, i.e. for the ontology traversal algorithm to identify atomic terms consistent with query terms. The choice of a multi-context, large ontology might have appeared unsuitable, although an examination of the unmatched *S+S OQE* terms revealed that few were related to the other contexts; therefore, the multi-context argument was not considered a critical factor on the basis of the *S+S OQE* results.

The decision to use a larger ontology context set provided for a comparison with T401 and T416. As indicated, the extent of T438 query expansion was confined to solely *S+S OQE*, unlike T401 (*All OQE*) and T416 (*S+S+R OQE*). Whilst the *S+S OQE* limitation may have affected possible P&R improvements, it did provide a means to compare the results between the different expansion approaches. Tourism contains a number of asserted conditions defining relationships between *specific* tourism concepts; however, relationships are also specified between tourism concepts and imported context concepts (e.g. ActivityHoliday involvesActivity *sport:*Mountaineering). If *S+S+R OQE* had been used, the APV results might have been improved but, conversely, precision could have been adversely affected by greater numbers of non-tourism specific terms, which might have been irrelevant to a query. In hindsight, the use of *S+S+R OQE* in T438 would have been an important factor in both determining the benefit of using a multi-context ontology and further validating the *S+S+R OQE* approach.

T438's topic statement provided a more general query objective ("Tourism and its increase"), compared to the more specific T401 and T416 topics; as a result, the encapsulation of "increase" to "country as a whole" presented a greater query formulation challenge; this was considered a key reason for the lower P&R results. Clearly, the nature of a query, allied with the suitability of an *OQE* context, must represent critical success factors.

From a recall perspective, the difference between (*optional*) Oo mode (90%) and Ko (74%) provides a justification for *OQE*. In contrast, (*must-have*) Km and Om modes achieved only 53% recall, suggesting no benefit in *must-have OQE*; although, the low recall was possibly because the *must-have* terms may have been "inappropriate" (given the less specific query relevance guidelines compared to T401 and T438). Again, the choice of *must-have* term could have excluded relevant documents, as opposed to simply relying on the *tf-idf* and weighting algorithm in *optional* mode.

Finally, as mentioned in the T401 and T416 reviews, the *OQE* results may have been improved if different weightings had been applied, as indicated by the subsequent T401 *S+S OQE* weight reversal experiments in subsection 4.4.3.

4.3.7 Reflections on Hypotheses

Comments are now provided for the hypotheses proposed in subsection 1.9.2.

Hypothesis (i) – "hierarchical (*S+S*) *OQE* can have a positive impact on precision and recall, although class hierarchy expansions alone may not produce optimal results. Query term-matched classes may have more beneficial wider semantic *relations* with other classes" (*S+S+R OQE*).

The experiment primarily shows that inconclusive results were achieved from the perspective that *OQE* failed to outperform keyword-only search. T438's *OQE* APV results demonstrated poor search effectiveness and failed to achieve the APV levels and successes identified in T401 & T416; given that, on average, only a third of all queries were improved by *OQE* and keyword mode was predominant in approximately 60% of queries. Nevertheless, the few *OQE* successes should not be ignored, as they could produce tangible search improvement in a bi-modal search engine process. On a secondary note, whilst the precision results were inconclusive, the group *optional* mode recall outcomes (Oo mode 90% versus Ko 74%) do favour *OQE*.

The T401 and T416 results have supported the "*extended OQE"* element of the hypothesis. The failure to conduct *S+S+R OQE* in T438 has highlighted a missed opportunity in this experiment; therefore, the effect of *S+S+R OQE* in T438 will be examined in subsection 4.4.4. However, overall, the *OQE* APV results have been poor and inconclusive; sufficient to refute the hypothesis, in terms of precision improvement.

Hypothesis (ii) - "higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes".

As mentioned in T401, this will be evaluated by comparing the initial T401 weightings and APV outcomes with revised T401 weighting experiment outcomes in subsections 4.4.1 to 4.4.3.

Hypothesis (iii) – "topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion, despite their potentially restrictive coverage, … as opposed to contextually wider, or more comprehensive, ontologies …. to avoid superfluous query expansion".

The T438 experiment used a wider, multi-context ontology (not designed for purpose but associated with tourism); compared to the T401 and T416 contexts, developed specifically for their query topics. T438 has produced inconclusive APV results, compared to T401 and T416, and the average *OQE* ratio of 4:16, with an 81% class matching, represented the lowest *OQE* ratio of the three experiments and suggested a less efficient *OQE.*

As mentioned in the critical review, the choice of a multi-context ontology might have appeared unsuitable but few unmatched *S+S OQE* terms were related to the other contexts; therefore, the multi-context characteristic was not considered relevant on the basis of solely *S+S OQE* results. In hindsight, the use of *S+S+R OQE* in T438 would have provided additional data to more effectively evaluate the use of small topic specific/self-contained ontology contexts against a multi-context ontology; this represented a missed opportunity to fully test the hypothesis. Therefore, the merit of using T438's wider set of ontology contexts will be further evaluated: by comparing T438's APV and class matching results (including queries using *S+S+R OQE*) against T401 and T416, in subsection 4.4.4.

4.4 FURTHER EXPERIMENTATION WITH T401 AND T438

After completion of the 3 main *OQE* experiments, additional data was generated for further evaluation of hypothesis (ii). A limited set of *S+S* and *S+S+R OQE* experiments were conducted using the T401 Immigration context (see subsections 4.4.1 to 4.4.3), by applying different combinations of concept relevance weightings compared to the weighting approach presented in subsection 3.2.5. Further data was also generated for evaluation of hypothesis (iii), including a comparison of small context *OQE* against a larger, more generalised ontology *OQE*, simulated by comparing some T401 Immigration results against results from an extended Immigration context, incorporating concepts from the SUMO ontology (see subsection 4.4.4).

Subsection 4.4.5 provides comments related to hypotheses (ii) and (iii).

4.4.1 Comparing Higher and Lower Term Relevance Weight APVs

The original relevance weighting approach (subsection 3.2.5) for a concept's semantic distance, from a query term matching class, was based on weights for the direct matching class, parent, relation and child classes, and individuals, being set at 1.0, 0.7, 0.5, 0.3 and 0.1 respectively. The main (T401, T416 and T438) experiments all used these query term relevance weightings.

As a preliminary experiment, the worst performing T401 query (Q10) was repeated using *optional S+S+R OQE* mode (Oro), to understand the effect of adopting varying combinations of weightings. Q10 was re-run using the weight sets (A, B and D) shown in Table 23. The class type headings, e.g. Parent [S] and Relation [R] are shown simply to reflect likely positions in the *S+S+R OQE*.

Table 23. Matrix of comparison class relevance weights.

Weight Set	Direct	Parent [S]	Relation [R]	Child [S]	Individual [S/R]
A	1.0	1.0	1.0	1.0	0.1
B	1.0	0.9	0.7	0.5	0.1
C	1.0	0.7	0.5	0.3	0.1
D	1.0	0.5	0.3	0.2	0.1

Weight set C represents the original weightings used for T401 Q10. Sets A, B and D retain the same direct class weight (1.0); set D uses lower weights for Parent, Child and Relation (i.e.

S+S+R), whilst sets A and B apply progressively higher *S+S+R* weights. Weight set A in effect removes weights in the *OQE*, as it treats all classes the same; this has the effect of making *S+S+R OQE* consistent with T401's *All OQE*, where all classes were weighted 1.0.

The Fig. 122 Oro P&R results compare the original Q10 set C weights to the results for the variant weight sets A, B and D; the P&R curve for *optional* keyword-only (Ko) is also shown for comparison purposes.

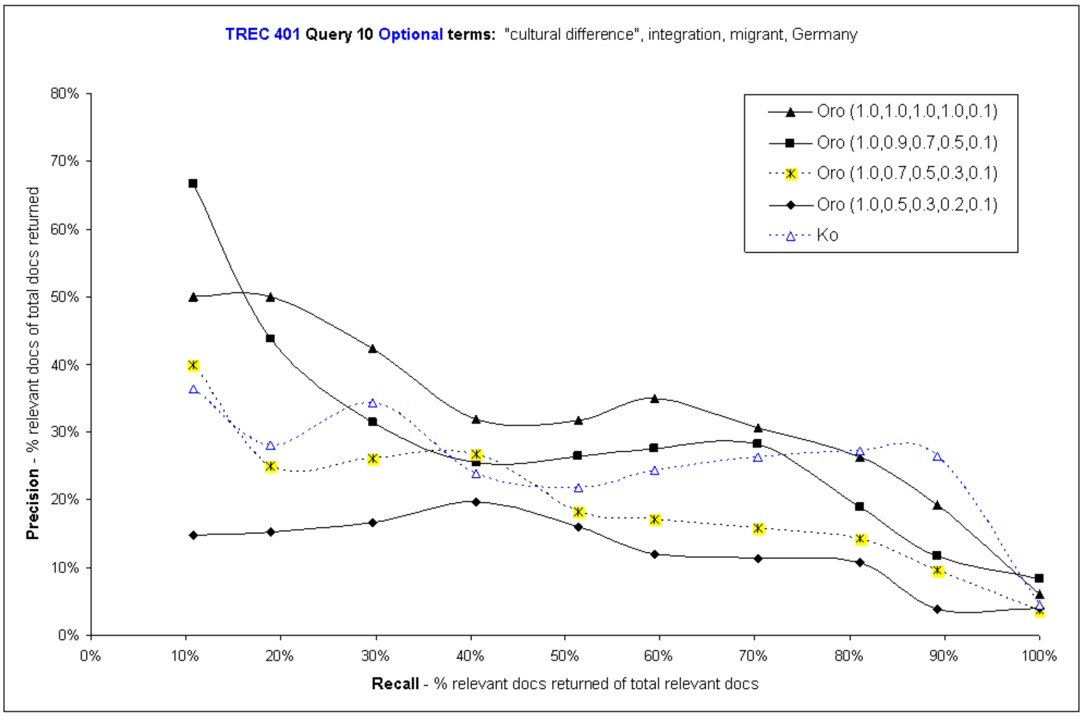


Fig. 122. P&R results based on matrix of relevance weights.

Using the Ko result as a base, the APV measures show that lower Oro weights result in a weaker precision curve and higher Oro weights improve precision. The APVs for the low weight set D and original weight set C are 16% and 30% respectively, whereas the measures for higher weight sets B and A are both 47%; therefore, lower weighted D and C performed badly, whilst A and B search effectiveness outcomes were very good compared to Ko. The result is interesting because Voorhees (Voorhees, 1994) found that assigning lower weights to query expansion concepts enhanced retrieval performance.

The APVs for set A (in effect a "non-weighted" approach), and B (raised weightings), suggest that even better *OQE* results could have been generated had they been applied in the three main experiments; this view is supported by the original T401 Qs 1-6 results, where a uniform 1.0 weighting in *All OQE* mode produced very good APVs over keyword.

4.4.2 APVs for Reversed Relevance Weights in *S+S+R OQE*

Based on the T401 Q10 results in the previous subsection, further modified *S+S+R OQE* weight experiments were conducted to measure the effect across the T401 Q7-10 group. The Oro mode was used again, this time to compare set A's uniform 1.0 weight (denoted in Fig. 123 as Non-Wtd Oro), the original set C weights (Std-Wtd Oro), and a modified set C with parent (0.7) and child (0.3) weights being *reversed* to 0.3 and 0.7 respectively (Rev-Wtd Oro); relation class weights were unchanged. The comparison P&Rs, including Ko, are shown in Fig. 123.

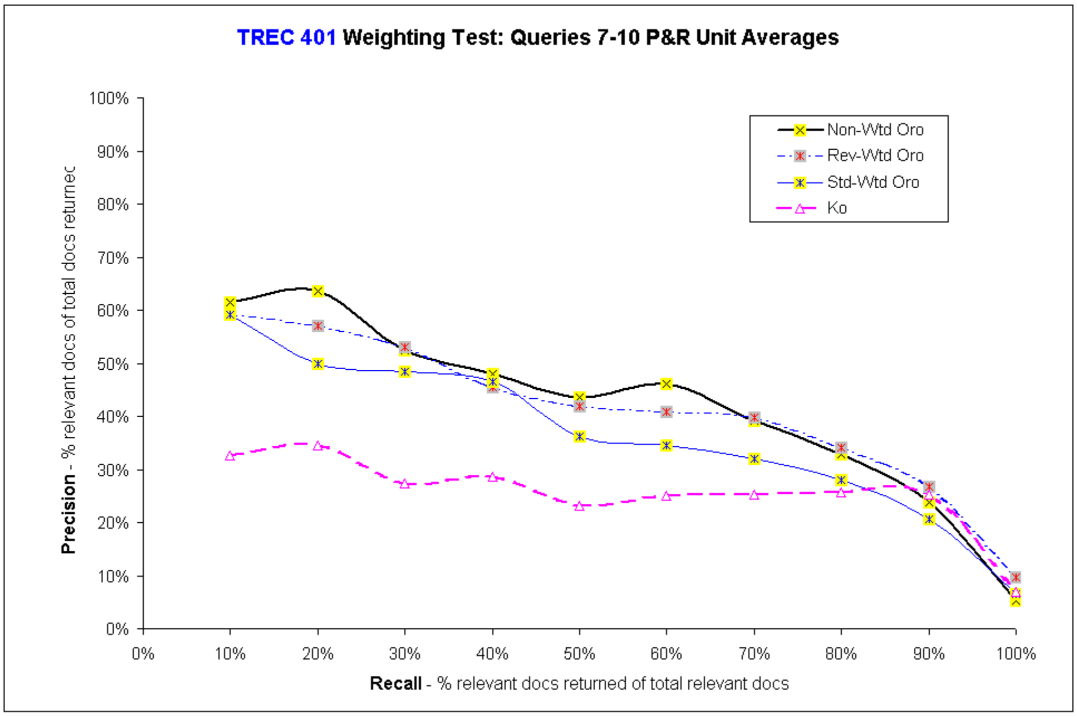


Fig. 123. P&R comparisons for Q7-10 Non-Wtd, Rev-Wtd and Std-Wtd *S+S+R OQE*.

The primary APV measures for the Q7-10 group were: Ko 32%; original (Std-Wtd Oro) set 53%; reversed parent and child set C (Rev-Wtd Oro) 56% and "non-weighted" set A (Non-Wtd Oro) 59%. Whilst Std-Wtd Oro was the least favourable *OQE*, all demonstrated good search effectiveness improvement over Ko, with Rev-Wtd and Non-Wtd achieving very good results against Ko. A secondary outcome was that higher precision was achieved between 10% and 70% recall with all Oro modes.

This result suggested two issues: firstly, that higher weightings for specialisation (sub class) concepts are potentially more appropriate and effective; but secondly, given the "non-weighted" performance, the application of relevance weightings, in any form, could be a less optimal and counter-productive strategy for identifying relevant terms and documents, i.e. query expansions should fully recognise the value of a wider *OQE*-related set of terms in the document by not differentiating between semantic relations.

4.4.3 APVs for Reversed and Exaggerated Weights in *S+S OQE*

Given that relation class weights were not modified in subsection 4.4.2, further query tests were made to examine the P&R impact when reversed and *exaggerated* weights are applied, this time in *optional S+S OQE* mode (Oo). Three sets of comparisons were made: Std-Wtd and Rev-Wtd (as in previous subsection) and an exaggeration of the sub class weight, i.e. from 0.3 to 1.7 and retention of the super class 0.7 weight (Extd-Wtd). The results for the combined Q1-10 group are shown in Fig. 124.

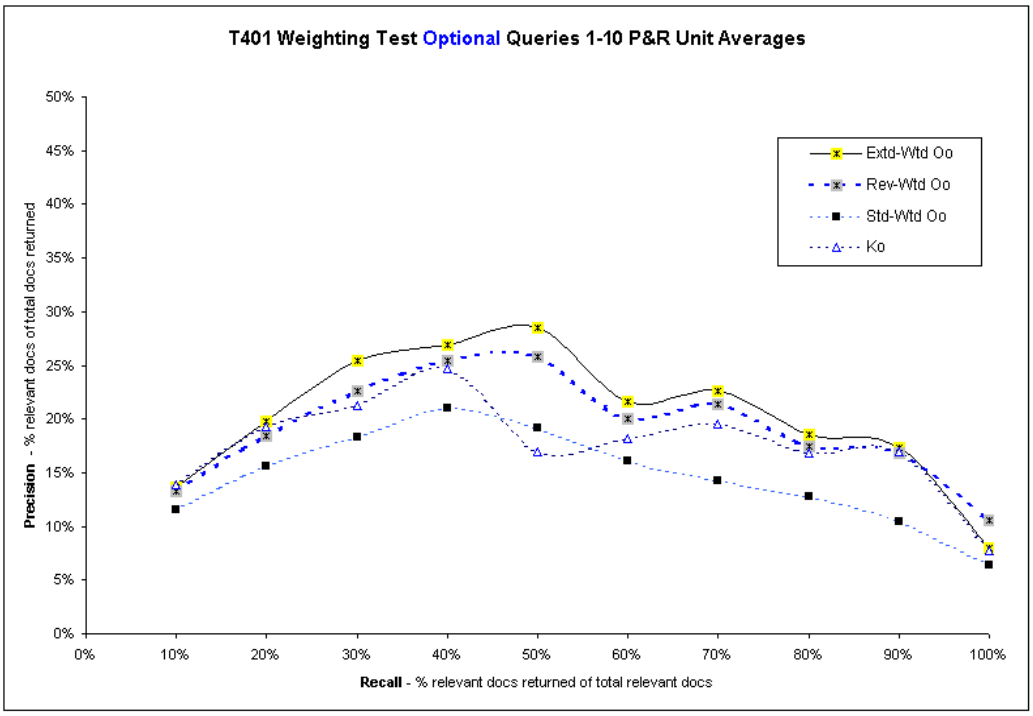


Fig. 124. P&R comparisons of *S+S OQE* using reversed and exaggerated weights.

The primary Q1-10 group APV measures were: Ko 18%; Std-Wtd Oo set 15%; Rev-Wtd Oo set 18% and exaggerated set (Extd-Wtd Oo) 20%. The Std-Wtd and Rev-Wtd Oro *OQEs* produced poor search effectiveness, with Std-Wtd falling below Ko and Rev-Wtd failing to better Ko. Extd-Wtd Oo was the most favourable *OQE* but achieved only a small/negligible improvement in search effectiveness. One reason for the results could have been that 5 of the 10 queries had very low query term expansions, which may have minimised the effect of weight changes across the whole query group.

4.4.4 Comparisons of Context *OQE* against Larger Ontology *OQE*

The following two experiments provide further data and observations for testing the validity of hypothesis (iii). The first experiment compares the P&R results of the flat T401 Immigration context against an expanded and more hierarchical T401; achieved by simply including SUMO class names into Immigration (e.g. sumo:EthnicGroup) to give a *hybrid* Immigration and SUMO ontology (T401+SUMO). More generalised terms (super classes), of Immigration classes, were identified in the SUMO ontology and then mapped to T401 Immigration classes. The second experiment provides an evaluation of P&R results using a more credible *OQE* comparison between the topic specific T401 and T416 contexts and the less topic specific, multi-context T438 ontology; this required new T438 P&R data, based on a set of *S+S+R OQE*s.

Creating a Larger T401 by Including SUMO with T401 Concepts

An extract of the contrived Immigration and SUMO ontology is shown, for schematic representation only, in Fig. 125. Immigration classes have been highlighted in blue to provide a demonstration of the effect of the mapping to just a small number of SUMO concepts. A clearer representation of the ontology is provided in Appendix C.

The ontology shows a number of child → parent mappings, where an Immigration class has been identified as the sub class of a SUMO class (X → sumo:Y). As a consequence, the number of classes has been increased from 41 to 124 (300%); examples of such mappings to their next generalisation levels can be seen more clearly in the Appendix C version, e.g. with Shelter → sumo:Structure and sumo:Construction, Migrant → sumo:Traveller and sumo:Traveler, Integration → sumo:GroupAction, Security → sumo:Fearlessness and EthnicMinority → sumo:EthnicGroup.

The T401 query matrix (Table 8, subsection 3.3.1) was re-used, this time with queries Q1-10 all based on *optional S+S OQE* queries (Oo). The T401 to T401+SUMO comparisons of generated *OQE* terms for the two ontologies are shown in Table 24.

Table 24. Comparison of T401 versus T401+SUMO *OQE* terms returned.

Query Set	T401 Oo *OQE*	T401+SUMO Oo *OQE*
Q1	7	20
Q2	7	18
Q3	12	34
Q4	7	23
Q5	6	28
Q6	6	15
Q7	8	33
Q8	10	31
Q9	5	16
Q10	13	31
Ave Q1-10	8	25

The T401+SUMO query expansion generated an average 25 query terms across all query sets compared to an average 8 terms for T401; the increase in term expansion (310%) was very similar to the difference between the two ontology class sizes. The resulting P&R profile comparisons for the merged query group Q1-10 are shown below in Fig. 126 and Fig. 127.

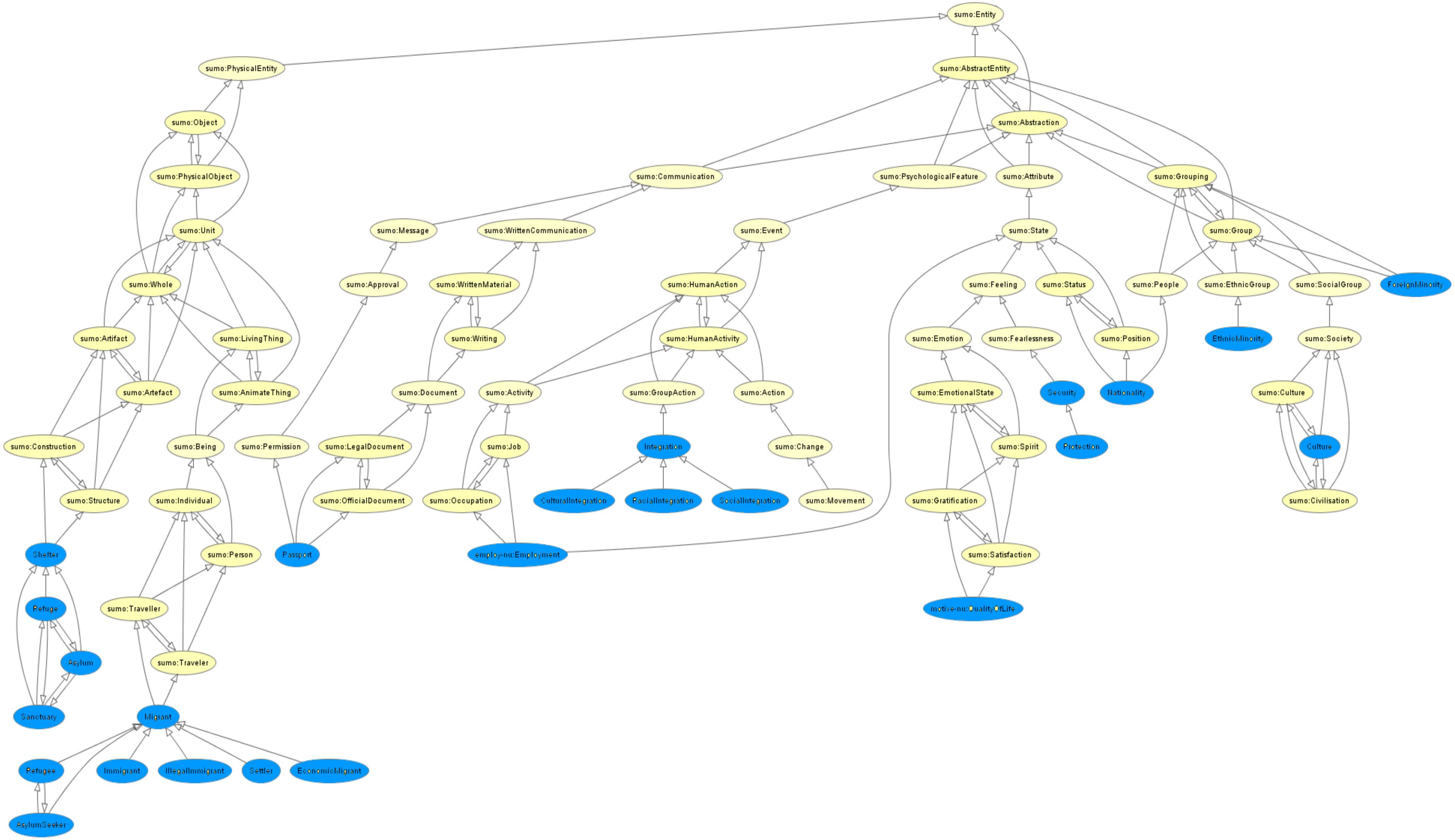


Fig. 125. Immigration classes mapped to SUMO. (for schematic representation only).

Comparison of T401 APVs against T401+SUMO APVs

A comparison of the average P&R profiles is shown in Fig. 126, based on average unit percentages, and Fig. 127, using the MEA-based approach. In Fig. 126, the T401-only expansion has produced better precision result over T401+SUMO, although the precision performance is weak overall. The primary APV measures show 14% for T401 Oo and 12% for T401+SUMO Oo. The Fig. 126 result is also reflected in Fig. 127, although MEA gave a much stronger precision curve for both.

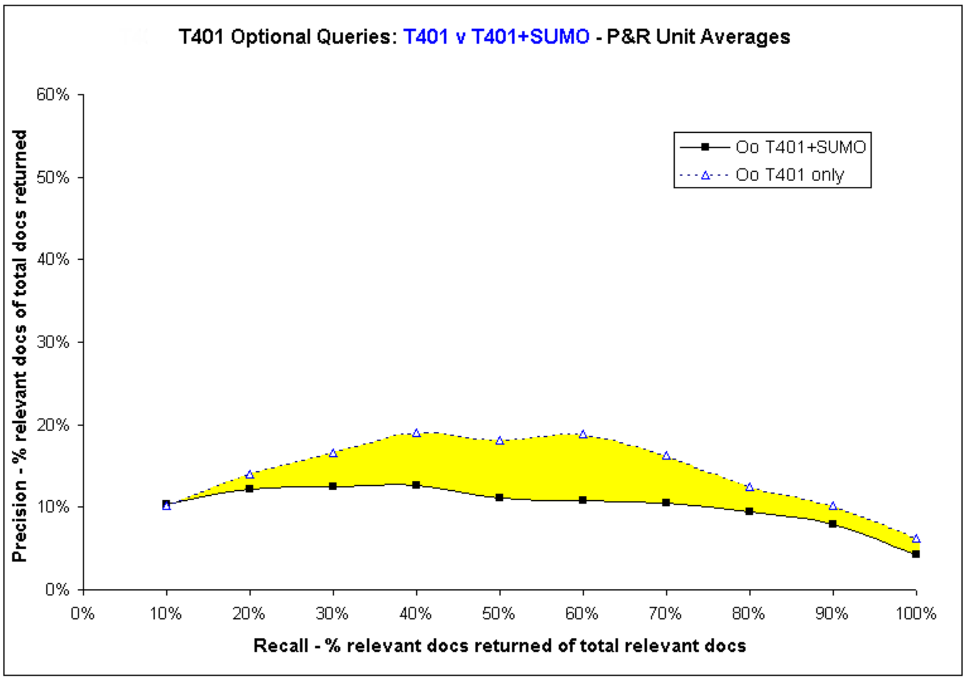


Fig. 126. Comparison of T401 with T401+SUMO.

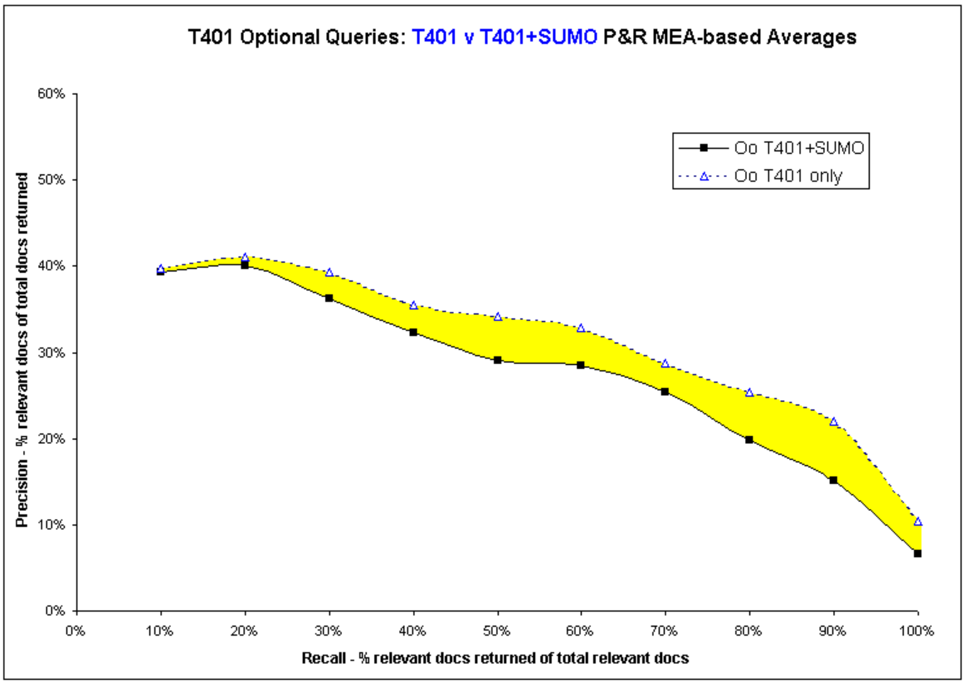


Fig. 127. MEA-based comparison of T401 to T401+SUMO.

Fig. 127's MEA-based outcomes were 40% for T401 Oo and 38% for T401+SUMO Oo. The APVs confirm a slight improvement in *OQE* search effectiveness when using T401 only. However, a secondary outcome is that T401 has performed better than T401+SUMO along the whole recall range in both graphs; suggesting better *OQE* results may be achieved with more specialised ontology contexts, i.e. avoiding generalisation levels of a class hierarchy.

Comparing T401 and T416 with T438

The benefit of using smaller topic specific *OQE*, as opposed to wider multi-context *OQE*, can perhaps be further considered by examining the trends shown between the group P&Rs, of the main T401, T416 and T438 experiments in Fig. 128, Fig. 129 and Fig. 130 respectively. It should be recalled that the T401 Fig. 128 Oo P&R curve represents a combination of *All OQE* and *S+S OQE*. It should be further noted that, for ease of graph interpretation (given the low T438 precision levels), Fig. 130 is presented with a precision scale of 0%-10%.

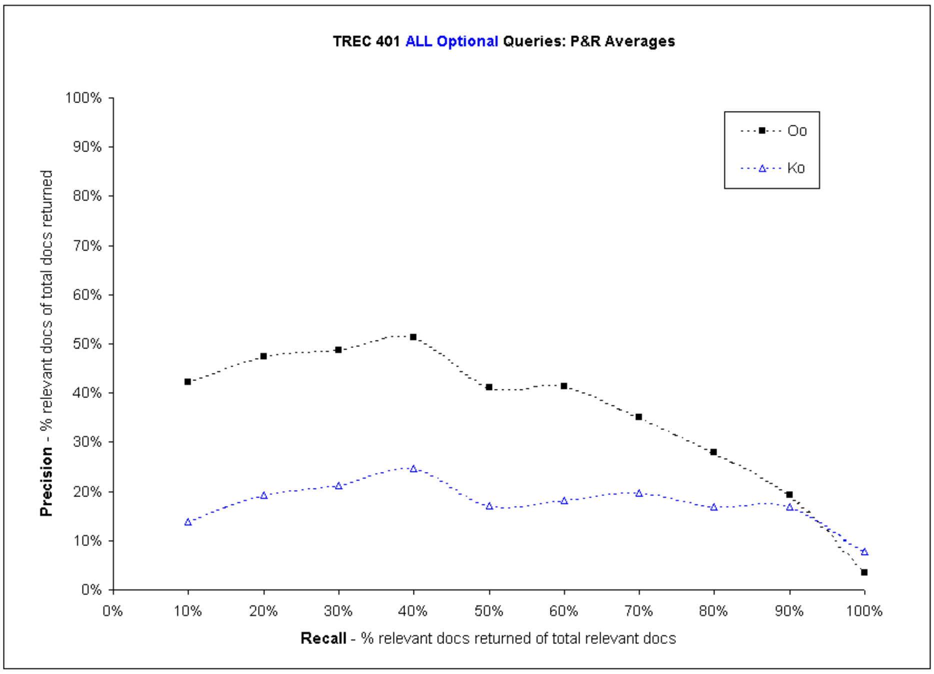


Fig. 128. T401 Overall P&R for *optional* queries.

In Fig. 128, the primary APV outcome for T401 was that Ko was 18% and Oo was 46%; this demonstrated a strong *OQE* precision performance and a good Oo search effectiveness improvement over Ko. A secondary outcome was that *OQE* delivered much improved precision between the 10-80% recall points.

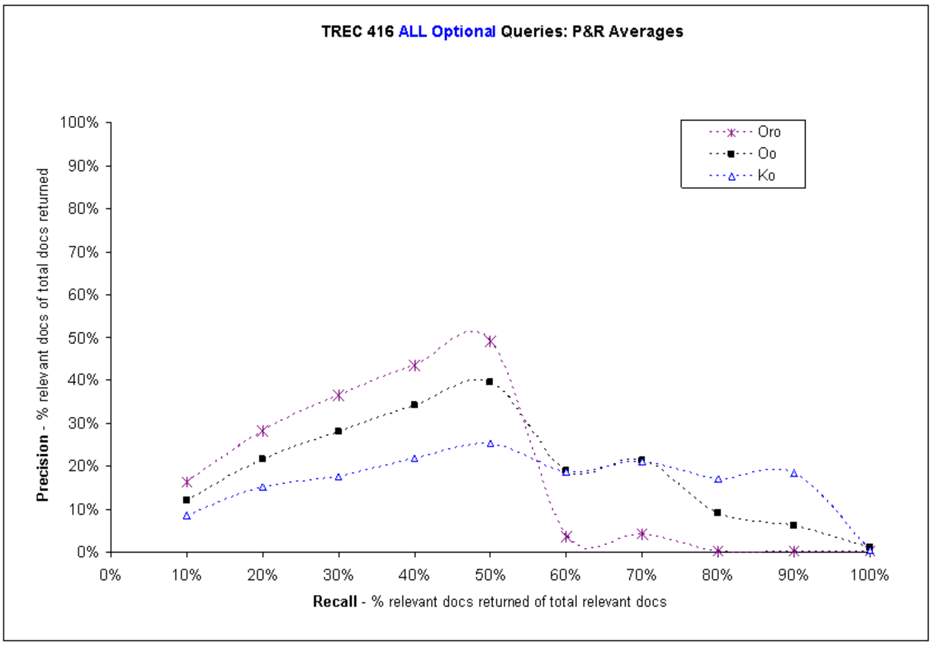


Fig. 129. T416 Overall P&R for *optional* queries.

In Fig. 129, the key APV outcomes for T416 were Ko at 14%, Oo 21% and Oro 27%; this again demonstrated the benefit of *OQE*, with *S+S+R OQE* predominant. *OQE* provided a good APV improvement in search effectiveness over Ko. A secondary *OQE* outcome was the improved precision between 10-50% recall.

In contrast, T438's query outcomes provided markedly lower level of precision than T401 and T416. Fig. 130 shows APV of only 2.1% for Ko and 2.6% for Oo. Given that Fig. 130 conflicted with the earlier MEA-based Fig. 106, *OQE* search effectiveness impact was considered both negligible and inconclusive compared to Ko.

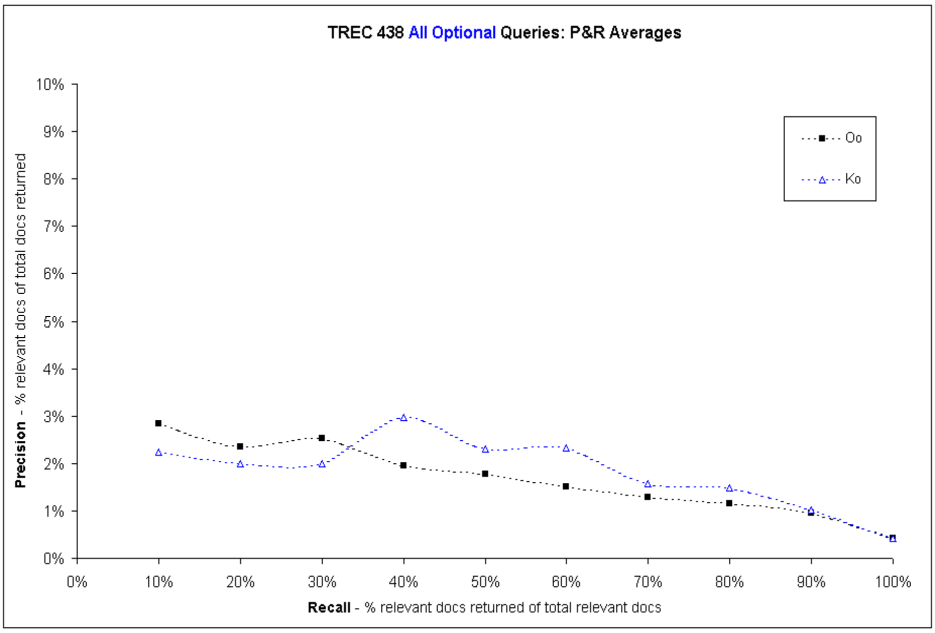


Fig. 130. T438 Overall P&R for *optional* queries.

The T438 experiment used a much larger ontology, not designed for purpose, with the highest precision rates (circa 6%) markedly lower than T401 and T416 (between 50% and 80%). In addition, the T438 average *OQE* of 13 of 16 terms being matched in document set (81% class matching), indicates a less efficient *OQE* and contribution towards P&R; these figures result from the lowest average *OQE* ratio of the three experiments, i.e. 4:16 vs. 4:28 for T401 (89% matching) and 4:21 for T416 (95% matching). Whilst the T401 *OQE* ratio should be put in perspective (it involved *All OQE* and was therefore less query term specific), the results appear to suggest that a smaller topic specific ontology context can produce better results than a wider multi-context ontology and that higher *OQE* ratios could result in improved class matching.

The inference in hypothesis (iii) is that the wider semantic links across a multi-context ontology could generate more non-query relevant terms in the query expansion, which could adversely affect precision. However, T438's subsection 4.3.7 hypothesis (iii) comments suggested further data was required to fully test the hypothesis; because, an important factor in justifying the use of small topic specific/self-contained ontology contexts, against a multi-context ontology, was a credible comparison with the extended *OQE* results of T401 and T416. Therefore, as T438 *S+S OQE* had generated few expansion terms from the other contexts contained in Tourism, a set of 13 *optional* mode queries were conducted using *S+S+R OQE*. Queries Q1-Q5, Q7, Q8, Q11, Q12, Q15-17 and Q19 were selected from T438's query matrix, for the experiment.

As mentioned previously, the Tourism ontology contained a number of asserted conditions defining relationships between *specific* tourism concepts and also between tourism concepts and imported context concepts, with the potential consequence that P&R could be either favourably or adversely affected, depending on the number of non-query relevant terms added to the expansion; *S+S+R OQE* allowed the effect of these relationships to be tested. The resulting P&Rs are shown in Fig. 131 and compare the combined query group results for Ko, Oo and Oro search options; the MEA-based comparison measure is shown in Fig. 132.

In Fig. 131, the APVs are: Ko at 2.1%, Oo at 3.5% and Oro 2.8%. Despite the low values, *OQE* has produced a modest search effectiveness improvement over Ko, with Oo unusually outperforming Oro.

In the MEA-based Fig. 132, APV levels are improved: Ko 6.4%, Oo 5.4% and Oro 4.6%. The MEA values show that both *OQE* modes performed badly - particularly Oro.

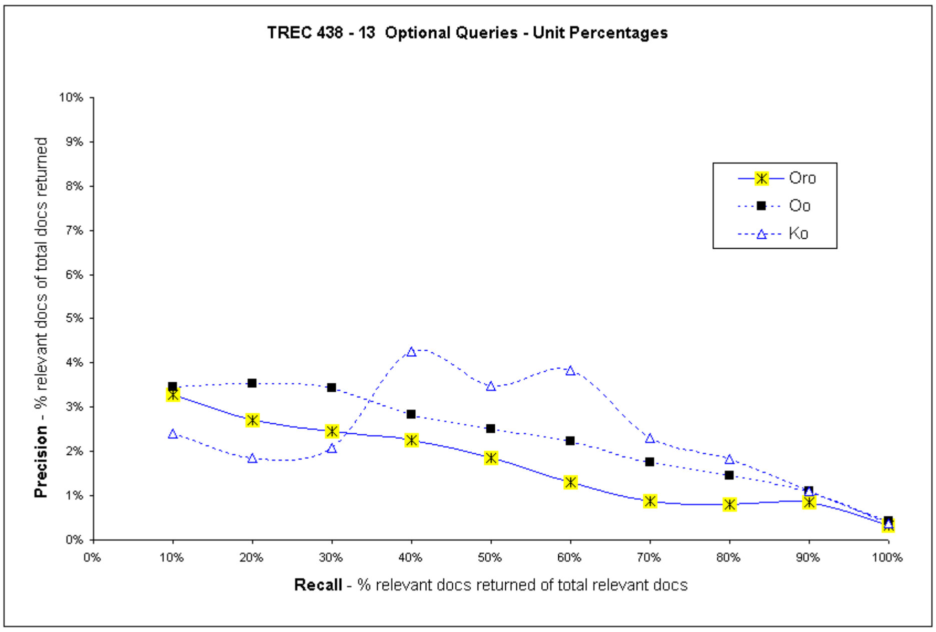


Fig. 131. Overall P&Rs for T438 Ko, Oo and Oro queries.

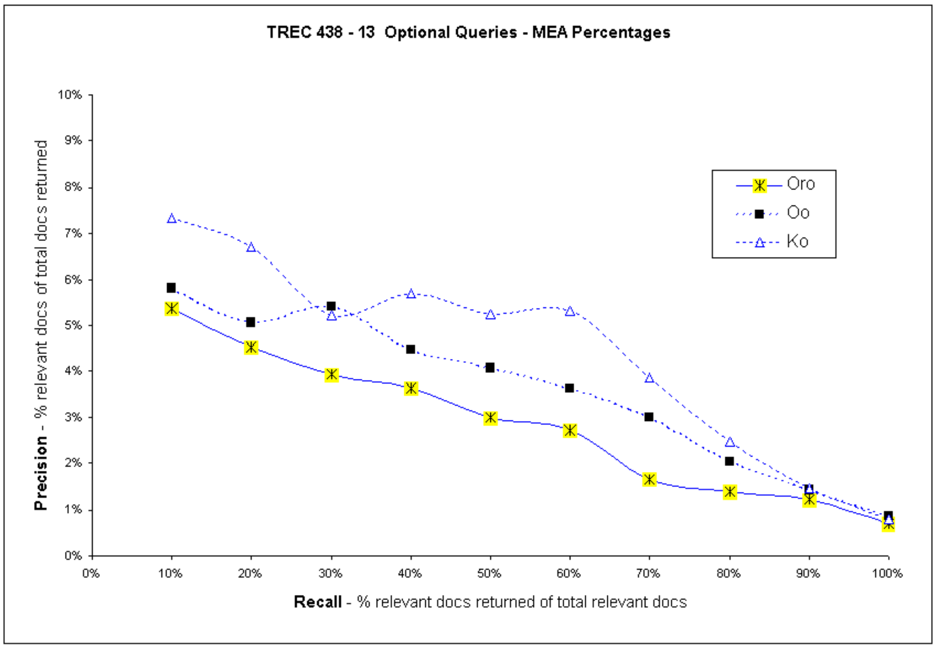


Fig. 132. Overall P&Rs for T438 Ko, Oo and Oro queries (MEA-based).

The results are inconclusive as the graphs present conflicting *OQE* APVs, showing either negligible or adverse search precision benefits. Clearly, there has been an adverse effect when using *S+S+R OQE* in a multi-context ontology. Secondary outcomes are that Oro has failed to enhance precision over Oo mode in both graphs and both *OQE* modes have been inferior to Ko, between 40% and 100%.

General Observations:

The expanded *OQE* results show that, unlike the improved precision results achieved with T401 (*All OQE*) and T416 (*S+S+R OQE*), T438 *S+S+R OQE* has produced poor precision outcomes.

To further emphasise the adverse effect of using a multi-context ontology for relation class expansion, the comparison class-matching statistics, with T438's *S+S* and *S+S+R OQE* shown separately, are shown in Table 25.

Table 25. Class matching comparison incorporating T438 *S+S+R OQE*.

Topic (*OQE Mode*)	Average *OQE* Ratio	Average *OQE* Terms	Average Matched Terms	Class Matching %
T401 (*All OQE*)	4:28	28	25	89%
T416 (*S+S+R OQE*)	4:21	21	20	95%
T438 (*S+S OQE*)	4:16	16	13	81%
T438 (*S+S+R OQE*)	4:37	37	28	76%

T438's *S+S+R OQE* ratio was the highest of the experiments but the number of classes matched was below its *S+S OQE* result (81%) and markedly lower than T401 (89%) and T416 (95%).

The APV and class matching results suggest that, compared to wider multi-context ontologies, topic specific/small ontology context *OQEs* can be highly search effective and can minimise the risk of potentially generalised and less relevant terms affecting precision, when using *S+S+R OQE*.

4.4.5 Reflections on Hypotheses

Comments are now provided for the relevant hypotheses proposed in subsection 1.9.2.

Hypothesis (ii) - "higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes".

The section 4.4 experiments demonstrated both the effect of different term relevance weightings on APV outcomes and the precision weakness of the original set C "standard" weights.

Fig. 122 (subsection 4.4.1) indicated that improved *OQE* results could have been generated in the three main TREC experiments, if the "non-weighted" (set A) and "raised weightings" (set B) had been applied; given that APVs for sets A and B were 47% compared to 30% for the original set C. The A/B weights gave very good improvements in *OQE* search effectiveness. The Fig. 123 (subsection 4.4.2) *OQEs* also highlighted positive findings: firstly, that higher sub class weightings may be potentially more appropriate and effective; secondly, the good "non-weighted" performance (59% APV versus 53% and 56% APVs for set C and reversed S+S set C respectively). Finally, in subsection 4.4.3, the Fig. 124 "reverse-weighted" (18% APV) and "exaggerated weighted" (20% APV) approaches both confirmed the benefit of emphasising the weight of a query term matched class's sub classes over its super classes.

However, whilst the APV outcomes confirmed good and further improved *OQE* search effectiveness, the subsection 4.4.1 and 4.4.2 outcomes suggest relevance weightings, in any form, could be a sub-optimal strategy for identifying relevant terms and ranking documents. The “non-weighted” approach in effect implies that *OQE*s should not differentiate between hierarchical and semantic relation terms.

In conclusion, the good APV results at first indicate that raised weightings and/or higher sub class weights can deliver improved APVs (10+%) and *OQE* search effectiveness over the original weightings; these outcomes support the hypothesis. However, the “non-weighted” APV performance (i.e. a *tf-idf* algorithm free of term relevance weightings), as also characterised by the results of T401 *All OQE* mode, in effect, refutes the hypothesis. Given the observations, more experimentation should be conducted, to further validate hypothesis (ii).

Hypothesis (iii) - “topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion, despite their potentially restrictive coverage, … as opposed to contextually wider, or more comprehensive, ontologies …. to avoid superfluous query expansion”.

It should be clarified that the hypothesis does not mean to specifically suggest the use of *All OQE*; it assumes any method of ontology traversal, e.g. *S+S* or *S+S+R OQE*. The experiments demonstrate that superfluous query expansion (attracting non-query relevant classes/terms), by *S+S OQE* (adding more generalised classes) and *S+S+R OQE* (adding relation classes from the wider multi-contexts in T438) can adversely affect P&R and, more importantly, APVs. The T401 versus T401+SUMO experiment addressed the more hierarchical *S+S OQE* aspect and the additional T438 experiment addressed the multi-context characteristic enabled through *S+S+R OQE* (subsection 4.4.4).

The T401 versus T401+SUMO graphs showed a small APV improvement when using T401 only, compared to T401+SUMO; this was reinforced by T401 performing better than T401+SUMO, along the whole recall range. The outcomes provide evidence to support the hypothesis that, by avoiding the generalisation levels of a class hierarchy, greater *OQE* search effectiveness can be achieved by using a specialised ontology context. The additional T438 experiment graphs showed that, in a wider, multi-context ontology, *S+S+R OQE* can adversely affect APV outcomes; unlike the improved APV results achieved with T401 (*All OQE*) and T416 (*S+S+R OQE*). The relationship class expansion was inferior to *S+S OQE* in both T438 graphs and an increased *OQE* ratio was the least effective in class matching.

In conclusion, the results of both experiments provide evidence to support hypothesis (iii).

5 EVALUATION OF T401, T416 & T438 EXPERIMENTS

As a preliminary and evaluation of precision performance, comparisons of query modes used in the three main query topic experiments are shown in the Fig. 133 line graph and Fig. 134 bar chart; they are derived from a "league" table of two-way results comparisons, i.e. for each of the Ko vs. Oo and Km vs. Om query sets. For each query set, the modes were awarded either 1 or 2 points, with 2 awarded for the highest precision at a specific recall point (i.e. for the 10%, 20% and 30% recall). The Oro and Orm search options were excluded for this comparison as they only applied to T438.

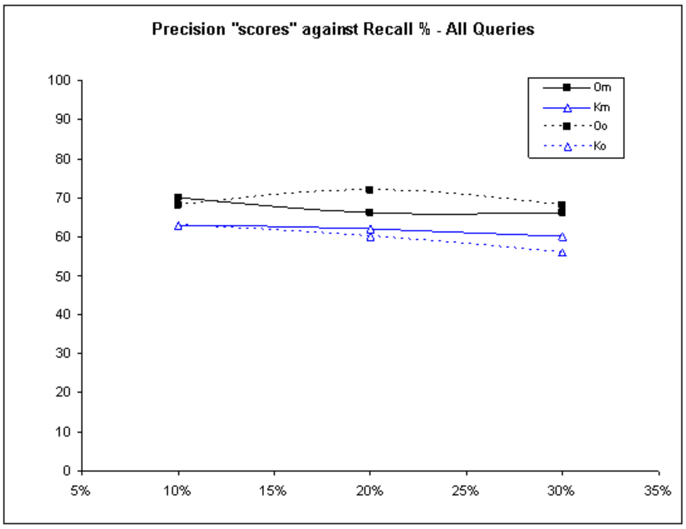


Fig. 133. Line graph comparison of Ko, Oo, Km and Om queries.

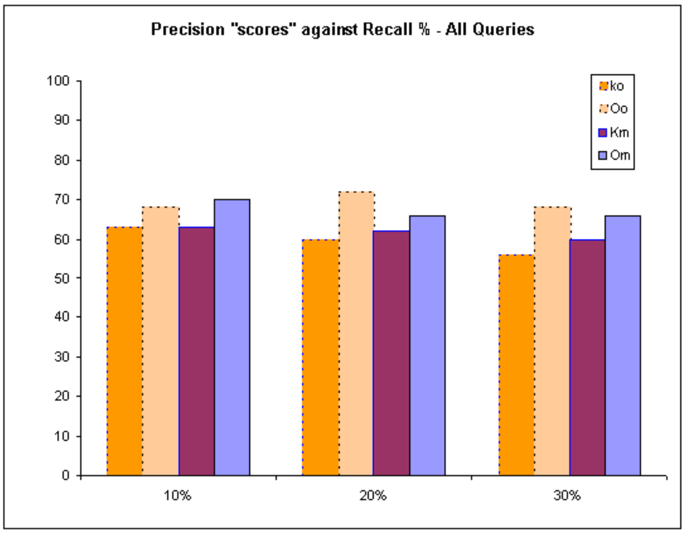


Fig. 134. Bar chart comparison of Ko, Oo, Km and Om queries.

The measures show that, across the experiments, Oo performed better than Ko and, Om performed better than Km; the greatest improvement was between Oo and Ko. It is emphasised that, even though Oo *appears* better than Om, the outputs do not imply that Oo *is* better than Om; the measures provide only a pairwise comparison, e.g. Ko vs. Oo.

5.1 SUMMARY OF EXPERIMENT RESULTS

Subsections 5.1.1 and 5.1.2 provide summary results for the main T401, T416 and T438 experiments. Subsection 5.1.3 provides comments regarding the additional experiments.

5.1.1 Performance Outcomes using APV Measures

As mentioned in subsection 3.1.7, to ensure a consistent performance evaluation of the 3 main experiments, query mode search effectiveness was primarily based on an average of the 10%, 20% and 30% recall point precision percentage values (the APV).

Specific individual query set APV outcomes have been presented in subsections 4.1.1, 4.1.2, 4.2.2 and 4.3.2. APVs for all 80 main experiment queries, i.e. 40 query sets in both *optional* and *must-have* modes, can be found in Appendix I. Table 26 provides a summary of the query mode successes, based on primary APV outcomes for the T401, T416 and T438 experiments.

Table 26. Comparisons of T401, T416 and T438 by query mode APV successes.

Optional Mode					
TREC Topic	No. of queries	% Ko Top	% Oo Top	% Oro Top	% Tied
T401	10	10%	90%	-	0%
T416	10	0%	10%	60%	30%
T438	20	65%	30%	-	5%

Must-have Mode					
TREC Topic	No. of queries	% Km Top	% Om Top	% Orm Top	% Tied
T401	10	10%	90%	-	0%
T416	10	10%	0%	50%	40%
T438	20	55%	35%	-	10%

The results summary shows that the two topic specific ontology contexts provide the most successful APV outcomes.

- T401 *All/S+S OQE* APVs were clearly the most search effective: in 90% of *optional* and *must-have* queries.
- T416 *S+S/S+S+R OQE* APVs were highly search effective compared to keyword only: in 70% of *optional* and 50% of *must-have* queries, with Oro and Orm predominant. Whilst 30%/40% of T416 queries showed tied outcomes, *OQE* performance was good overall and demonstrated the benefit of *S+S+R OQE*.
- T438 *S+S OQE*s presented a much weaker APV outcomes (30%/35%) compared to keyword (65%/55%); therefore, search effectiveness was poor. Table 26 does not

include the additional T438 *S+S+R OQE* experiments (subsection 4.4.4), where Oro performed badly; producing an adverse APV outcome when used in the multi-context ontology and failing to enhance precision over Oo mode.

5.1.2 Precision Successes and Recall Outcomes

In comparison to Table 26 above, Table 27 summarises the percentage of times each query mode produced the highest precision successes at 10, 20 and 30% recall points and the recall performances overall.

The results reflect similar outcomes to the APV performance in subsection 5.1.1 and confirm the benefit of extended (*All/S+S+R*) *OQE*: the most successful were T401's Oo, which includes *All OQE*, and T438's Oro. With the exception of T401, the results also show that *OQE* can produce improved recall, with Oro achieving 100% in T416 and Oo showing a marked increase to 90% over Ko in T438. Again, this is clear evidence that *OQE* can produce improved search effectiveness, in both precision and recall.

Table 27. Comparisons of T401, T416 and T438 query mode successes.

Optional Mode						
TREC Topic	Precision Successes			% Recall Achieved		
	Ko	Oo	Oro	Ko	Oo	Oro
T401	13%	77%	-	100%	100%	-
T416	0%	10%	67%	95%	95%	100%
T438	53%	43%	-	74%	90%	-

Must-have Mode						
TREC Topic	Precision Successes			% Recall Achieved		
	Km	Om	Orm	Km	Om	Orm
T401	13%	83%	-	100%	100%	-
T416	10%	0%	50%	81%	81%	81%
T438	50%	32%	-	53%	53%	-

The results are also dependent on query modes used: *optional* mode *OQE*s have been more search effective than *must-have* mode *OQE*, when compared to their respective keyword-only modes; *must-have* modes have resulted in overall weaker recall performance, for all query modes. However, a straight comparison between *optional* and *must-have* modes is not particularly important. *Optional* mode places full reliance on the *tf-idf* and term relevance weighting algorithms. *Must-have* can conflict with algorithm by restricting return of potentially relevant documents; this has been reflected in the poorer recall performances, where a *must-*

have term may have been a questionable choice. Therefore, the *optional* query mode results (Ko, Oo and Oro) are considered more relevant.

The form of ontology context was important; better results were achieved using ontology contexts that were specifically developed for the query topic, i.e. T401 and T416 versus T438, with multiple contexts interfacing with Tourism. There were clear issues with the T438 ontology: the ontology was not designed for purpose, multiple contexts were imported, and query term selection proved difficult as the precise nature of the query parameters were less specific in the T438 topic narrative. Flat ontology hierarchies can restrict *S+S OQE* results; the options are to consider using whole ontology for *OQE* (as in T401) or *S+S+R OQE* (as applied in T416).

5.1.3 Additional Experiments

As discussed in the additional experiments in section 4.4, the results of the three main experiments did not in fact fully demonstrate the potential of *OQE* because the initial term relevance weights tended to present *OQE* in an almost "worst-case" scenario. The main experiments were based on the original (standard) relevance weightings, e.g. with lower weights for sub classes, whereas the further *OQE* experiments used different weight combinations, e.g. by increasing all weights by 0.2 or by "removing" weights by applying a weight of 1.0 for all concepts. It was also found that reversing the super class and sub class weights (0.7 ↔ 0.3) also had a positive precision effect. The raised weight changes resulted in improved *S+S+R OQE* APV outcomes and good search effectiveness, as shown in subsections 4.4.1 and 4.4.2. Therefore, the performance measures could have demonstrated further improvements, with either a higher weighted or a *non-weighted* value in the *tf-idf* calculation.

5.2 CRITICAL REVIEW

The following points represent a critical review of the work undertaken.

i. The outcomes from the APV (5.1.1) and P&R (5.1.2) subsections, generated by *tf-idf* relevance scoring, have clearly demonstrated the success of *OQE* in improving search effectiveness using small ontology contexts. APVs have been used to provide a consistent and primary measure of performance and the focus on early recall points has also served to recognise that a typical Web user might only be interested in examining the first few pages of search engine results. The P&R outcomes helped to verify the APVs and identify a secondary benefit, i.e. *All OQE* and *S+S+R OQE* modes markedly improving precision in the 10-50% recall range.

   The T416 results show *S+S+R OQE* can achieve a higher recall than both *K* and *S+S OQE*; also that extending *OQE* beyond the subsumption relationship, by exploiting the wider semantic relationships between ontology classes, has been justified. The T401 results also demonstrated more favourable results when using *All OQE*; however, whilst the T401 and T416 results tended to favour *extended OQE*, it was acknowledged in the T401 and T438 critical review comments (subsections 4.1.7 and 4.3.6 respectively) that *S+S+R OQE* should have been used in all experiments; this highlights a missed opportunity to further validate the benefit of *S+S+R OQE* to test hypothesis (i).

ii. The high *OQE* ratios and levels of term matching (88%-95%), highlighted in subsection 4.4.4), demonstrate that the small T401 and T416 ontology contexts were very relevant to the experiments; however, whilst small modules can produce good results, flat hierarchies can adversely affect basic *S+S OQE*, e.g. T401 *S+S OQE* averaged 8 terms versus *All OQE*'s 41 terms. Similarly, T416 *S+S OQE* was 8 terms versus 21 using *S+S+R*. In retrospect, the use of small contexts, in T401 and T416, should have flagged the need for *S+S+R OQE* earlier.

iii. In the T401 and T416 experiments, keyword-only mode produced better precision results in the 10-30% recall range, in approximately 8% (3/40) of queries; in T438 the precision success was much greater - 53% (21/40) of queries – and the average across all three experiments was 30% (24/80). Therefore, would *OQE* alone be a justified approach? Based on the original relevance weightings used in the main experiments, the results would appear to suggest that, whilst *OQE* can make a positive impact, it is not a solution to replace keyword-only query, i.e. one or the other is probably an important supplemental solution in the search engine process. However, this view may be premature, given the precision outcomes demonstrated in the term relevance weighting experiments in subsections 4.4.1 to 4.4.3.

iv. The ontology modules proved useful with adaptive text input of query contexts and query term selection. Further, their small and specialised contexts may help to limit superfluous *OQE*, given their tighter relevance.

v. The process does not require creation of triple stores and there is no reason why, with procedural changes to data access, the process of using Semantic Web languages to interrogate traditional (unstructured) Web documents, could not be tested more widely on existing indexed databases, i.e. used in a more formal search engine environment.

vi. The process provided a contrived set of ontology contexts for use in predetermined query topics. A fully operational search engine would clearly require a vast increase in ontology contexts for many query topics; however, this represents challenge of scaling, as opposed to the merit of *OQE*, as a principle.

vii. The tool did not use the Ontology API inferencing capability (see section 2.1) to distinguish between asserted and inferred types; this would provide additional *OQE* capability. The requirement to run a classifier during ontology development might have been unnecessary if it had been decided to use the Jena Ontology API inferencing capability in SemSeT, i.e. to find inferred relationships – see "Ontology Traversal Example" in subsection 3.2.2.

viii. The subsequent experiments using modified term relevance weights (higher weightings and removal of weight differentials) suggest the three main experiment's *OQE* results were understated and presented a "worst-case" scenario; therefore, a valid criticism would be that term relevance weight testing should have been carried out beforehand, for the changes to substantively reinforce the research findings.

ix. The following comments refer to the research hypotheses proposed in subsection 1.9.2.

Hypothesis (i) – "hierarchical (*S+S*) *OQE* can have a positive impact on precision and recall, although class hierarchy expansions alone may not produce optimal results. Query term-matched classes may have more beneficial and wider semantic *relations* with other classes" (*S+S+R OQE*).

Subsection 5.1.1's APV summary highlighted strong *OQE* performance by topic specific contexts: T401 Immigration *All/S+S OQE* APVs produced the most search effective outcomes in 90% of *optional* and *must-have* queries; similarly, T416 Hydro-electric *S+S/S+S+R OQE* APVs provided good search effectiveness improvement in 70% of *optional* and 50% of *must-have* queries, with Oro and Orm predominant. Overall, T401 and T416 *OQE*s had the effect of more than doubling APV performances and, as a secondary benefit, have maintained the precision improvement differential up to the 50% recall range.

In contrast, the wider/multi-context T438 Tourism ontology *S+S OQE* APVs presented weaker outcomes: (30%/35%) compared to keyword (65%/55%). Overall, T438 search effectiveness was poor; further, the additional T438 queries, in subsection 4.4.4, also produced some adverse outcomes. Nevertheless, *OQE* successes in a third of queries not be ignored; they could offer incremental search benefits in a bi-modal search.

Clearly, applying *All/S+S+R OQE* on topic specific contexts as opposed to wider, multi-context ontologies differentiates *OQE* search effectiveness. Whilst, it might have been beneficial to conduct more extensive *S+S+R OQE*, the T401 and T416 *OQE* APV outcomes demonstrated the benefits of extending query expansion beyond *S+S OQE*, on topic specific contexts. On a secondary note, T416 and T438 *optional* mode recall outcomes were improved using *OQE*.

The above outcomes have provided strong evidence to support the hypothesis.

Hypothesis (ii) - "higher and more accurate document relevance scores (to improve precision) can be achieved by applying a simple relevance weighting system to query term-matched classes".

The subsection 4.4.1 to 4.4.3 experiments have demonstrated that "all-round" higher relevance weights and higher sub class weights have markedly improved (10+%) APV results, compared to the original weightings; these outcomes provide evidence to support the hypothesis. The revised weighting outcomes also suggest that improved *OQE* results could have been generated in the original main TREC experiments. At the same time, we have also learned that the "non-weighted" approach, as also characterised by the original T401 *All OQE* APVs, provides evidence to, in effect, refute the hypothesis.

Additional experimentation would be beneficial, to further test the hypothesis; however, the weighting experiments have provided good solutions for improving *OQE* precision.

Hypothesis (iii) – "topic specific or self-contained small ontology contexts can be highly effective for *OQE* expansion, despite their potentially restrictive coverage, … as opposed to contextually wider, or more comprehensive, ontologies …. to avoid superfluous query expansion".

The *OQE* experiments have clearly shown that the topic specific T401 Immigration and T416 Hydro-electric contexts provided the most successful APV outcomes. In contrast, the multi-context T438 Tourism ontology presented much weaker APVs and search effectiveness.

The additional subsection 4.4.4 T401+SUMO and T438 experiments were based on maximising the extent of ontology traversal and *OQE*; the APV results indicated that more hierarchical and/or wider ontologies have the potential to adversely affect precision, by adding less relevant terms to the query expansion.

Given the above outcomes, the experiments have provided good evidence to support the hypothesis.

x. Evaluation of the hypotheses was considered dependent upon answers to the following questions; these are provided below.

a) Has an impartial and unbiased search comparison process been employed?

This has been satisfied by using independent TREC data and query topics that were each supported by a query requirement, in the form of a topic statement, and a set of query relevance judgements (see sections 2.1 and 3.3).

b) Does the search tool support ontology traversal and relevance ranking mechanisms effectively and reliably?

The *OQE* tool was extensively tested to validate the integrity of: the ontology traversal (sub, super, equivalent and intersection classes); the *tf-idf* algorithm (matched term and document statistics stored for retrieval by the relevance ranking algorithm) and P&R data outputs. Both white-box testing (i.e. using test inputs during development, to verify paths through the code) and black-box testing (firstly, involving a small document control set with predetermined outcomes and secondly, using sample TREC data) were conducted (see subsections 2.2.3 and 3.1.2).

c) How useful were ontology query contexts, e.g. concept usage?

The high levels of extended *OQE* term matching (89%-95%) and corresponding P&R results show the smaller T401 and T416 ontology contexts were very relevant to the experiments, whereas the multi-context T438 extended *OQE* term matching was much lower (76%) and was reflected in the resulting P&R performance.

d) Did the results show meaningful improvements in either precision or recall?

Good precision improvement was evident in the T401 and T416 experiments.

xi. Understanding the issues of structural and semantic heterogeneity provided a helpful perspective upon which to understand the issues of semantics in IR.

6 CONCLUSIONS

The purpose of this PhD was to use *OQE* to improve search effectiveness by increasing search precision, i.e. retrieving relevant documents in the topmost ranked positions in a returned document list. The research experiments required a novel search tool to combine Semantic Web technologies in an otherwise traditional IR process using a Web document collection.

The above objectives have been successfully combined and the following conclusions provide an overview of results achieved, with hopefully an open, objective assessment of solutions presented; including identified success areas, the problems encountered and further solutions proposed.

6.1 HOW SUCCESSFUL – IN WHAT WAY

The experiments have successfully demonstrated that a process combining next generation Semantic Web languages, *OQE* and ordinary Web document information retrieval, can exploit the benefits of ontology semantics in a traditional search environment; i.e. without resorting to indexing RDF triple repositories and semantic reasoning-based RDF query languages.

The *OQE* experiment outcomes have justified the approaches adopted. The subsection 5.1.1 results summary has shown that the T401 and T416 contexts provided very successful APV outcomes: T401 *OQE*s were the most search effective in 90% of all queries; T416 *OQE*s were the most search effective in 70% of *optional* and 50% of *must-have* queries, with Oro and Orm predominant. The results also highlighted that the wider/multi-context T438 Tourism ontology *OQE*s were problematic: with weaker APV outcomes, resulting in successful *OQE*s in only a third of queries, and adverse T438 *S+S+R OQE* outcomes shown in subsection 4.4.4.

The hypotheses have been fully considered, in relation to the experiment outcomes, in critical review section 5.2. The additional section 4.4 experiments provided a larger results base for validating the hypotheses further, e.g. the different weighting experiments provided beneficial solutions for improving *OQE* APVs (by 10+%) and indicated that the original T401, T416 and T438 *OQE* results might have been understated. The experiments have provided good evidence to largely support the hypotheses; the only exception was where APV performance was improved by the "non-weighted" approach, i.e. this particular weight variation, in effect, refuted hypothesis (ii). Nevertheless, the weighting experiments provided useful solutions for improving *OQE* precision.

The SemSeT *OQE* engine has successfully achieved the primary objective of raising APV outcomes and improving search effectiveness: overall, T401 and T416 *OQE*s had the effect of more than doubling APV performances. In terms of secondary benefits, T401 and T416 *OQE*s maintained the (APV) precision improvement differential up to the 50% recall point and the T416 and T438 *optional* mode *OQE*s increased recall, by between 5 and 15 percentage points.

We have learned that topic specific context-based *OQE* is worthwhile: overall, the best APV results can be achieved by using ontology contexts (specifically relevant to the query topic) and extended *OQE*; as shown by the context-wide (*All OQE*) T401 Immigration approach and the hierarchical plus relation class expansion approach (*S+S+R OQE*) used in T416 Hydro-electric. Given that the experiments also revealed that *S+S+R OQE* used on a wider, multi-context ontology could produce less favourable results, the application of *All/S+S+R OQE* on topic specific contexts differentiates *OQE* search effectiveness. Finally, given the high percentage (89%-95%) of *OQE* terms matched in documents, the contexts proved to be very topic relevant.

6.2 PROBLEMS IDENTIFIED

i. As highlighted in subsection 3.2.4 (Extended Relation Class Algorithm), ontology traversal did not cater for all situations, e.g. when attempting to read union-based asserted conditions; any assertion had to be specified separately, as a solution was not found to list union operands using Jena Ontology API methods.

ii. Small ontology contexts with flat hierarchies may only provide limited potential for basic *S+S OQE* to deliver query term expansion; therefore, these ontology contexts require greater expressivity (via asserted conditions) to support wider or extended *OQE*, e.g. using relation class expansions.

iii. The section 4.4 experiments highlighted several options for improving the *tf-idf* relevance algorithm and presented seemingly conflicting choices: concept relevance weights need to be either generally higher, super and sub class weights should be reversed or relevance weight differentials should be removed.

6.3 SOLUTIONS PROPOSED

There is scope for further work, which could be directed in the following areas.

i. Further data from *S+S+R OQE* experimentation would be beneficial, to more fully consider hypothesis (i). Similarly, more extensive term relevance weighting experiments should be conducted, to further validate hypothesis (iii).

ii. Further *OQE* algorithm refinement, including traversal approach (e.g. inverseOf, partOf) and splitting any compound ontology terms identified during *OQE*, e.g. Cultural Integration or Tour Operator. However, this has the potential to generate more general terms. Similarly, SemSeT requires specific query terms to be entered in separate input boxes; the query functionality could be improved by allowing more complex queries to be input and using text analysis to reduce a natural language sentence (long tail) query into query term (short tail) sets for *OQE*, i.e. matching long tail elements with concepts.

iii. Precision and recall could be improved by identifying document annotations/metadata to recognise only contextually relevant documents, e.g. (Mika, 2008), and/or the development of Web document context metadata for ontology context matching; this could be related to a method/implementation for semantic indexing of documents, e.g. to speed up *OQE* processing in SemSeT.

iv. A means whereby the selection of a query topic can be used to facilitate automated integration of ontology contexts; when *OQE* needs to be applied on a wider application ontology, e.g. a query about “UK Travel” might require atomic/independent road, rail, air and population group contexts to be combined to facilitate ontology traversal.

v. The experiments have focused on topic-specific contexts, involving shallow structures; the design has provided an artificial ontology traversal control, which would not be available in a larger ontology. Therefore, experimentation to control the ontology traversal algorithm should be considered, to determine the optimum number of concept levels that would be included, e.g. in a sub and super class *OQE*; both from a query term relevance perspective and relative to the depth of hierarchy, when using a more extensive (wider ranging or more hierarchical) ontology.

vi. A methodology for developing a library of lightweight ontology contexts, based on modularised concepts best practise and the modular context approach demonstrated in section 1.7. The first step could be to prioritise contexts, by determining the most common query topics, and an analysis of Google’s query term “Suggest” functionality could provide an indicator.

REFERENCES

ARCH-INT, N. & SOPHATSATHIT, P. (2003) A Semantic Information Gathering Approach for Heterogeneous Information Sources on WWW. *Journal of Information Science,* **29**(5), pp. 357-374.

ASHBURNER, M., BALL, C. A., BLAKE, J. A., BOTSTEIN, D., BUTLER, H., CHERRY, J. M., DAVIS, A. P., DOLINSKI, K., DWIGHT, S. S., EPPIG, J. T., HARRIS, M. A., HILL, D. P., ISSEL-TARVER, L., KASARSKIS, A., LEWIS, S., MATESE, J. C., RICHARDSON, J. E., RINGWALD, M., RUBIN, G. M. & SHERLOCK, G. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics,* **25**(1), pp. 25-29.

BAADER, F., HORROCKS, I. & SATTLER, U. (2003) *Description Logics as Ontology Languages for the Semantic Web* [online]. Springer. Available from: http://www.cs.man.ac.uk/%7Ehorrocks/Publications/download/2003/BaHS03.pdf. [Accessed 6th November 2004].

BANSLER, J. P., DAMSGAARD, J., SCHEEPERS, R., HAVN, E. & THOMMESEN, J. (2000) Corporate Intranet Implementation: Managing Emergent Technologies and Organizational Practices. *Journal of the Association for Information Systems,* **1**(10), pp. 1-41.

BAR, F., KANE, N. & SIMARD, C. (2000) Digital Networks and Organizational Change: The Evolutionary Deployment of Corporate Information Infrastructure. In: *Proceedings of International Sunbelt Social Network Conference. Vancouver, British Columbia, April 13-16 2000*.

BATINI, C., LENZERINI, M. & NAVATHE, S. B. (1986) A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys,* **18**(4), pp. 323-364.

BECHHOFER, S., BROEKSTRA, J., DECKER, S., ERDMANN, M., FENSEL, D., GOBLE, C., VAN HARMELEN, F., HORROCKS, I., KLEIN, M., MCGUINNESS, D. L., MOTTA, E., PATEL-SCHNEIDER, P., STAAB, S. & STUDER, R. (2000) *An informal description of Standard OIL and Instance OIL* [online]. On-To-Knowledge. Available from: http://www.ontoknowledge.org/oil/downl/oil-whitepaper.pdf. [Accessed 12 January 2005].

BECHHOFER, S., GOBLE, C. & HORROCKS, I. (2001) *DAML+OIL is not Enough* [online]. University of Manchester, Information Management Group. Available from: http://citeseer.ist.psu.edu/448137.html. [Accessed 30 October 2004].

BECHHOFER, S. & HORROCKS, I. (2000) Driving User Interfaces from FaCT. In: *Proceedings of International Workshop on Description Logics (DL 2000). RWTH Aachen, Germany, 17-19 August 2000*. CEUR-WS, pp. 45-54.

BELLIFEMINE, F., POGGI, A. & RIMASSI, G. (1999) JADE: A FIPA-compliant agent framework. In: *Proceedings of Practical Applications of Intelligent Agents and Multi-Agents. April 1999*. pp. 97-108.

BENJAMINS, V. R., DAVIES, J., BAEZA-YATES, R., MIKA, P., ZARAGOZA, H., GREAVES, M., GÓMEZ-PÉREZ, J. M., CONTRERAS, J., DOMINGUE, J. & FENSEL, D. (2008) Near-Term Prospects for Semantic Technologies. *IEEE Intelligent Systems,* **23**(1), pp. 76-88.

BERGAMASCHI, S., CASTANO, S. & VINCINI, M. (1999) Semantic Integration of Semistructured and Structured Data Sources. *ACM SIGMOD Record - Special Issue on Semantic Interoperability in Global Information,* **28**(1), pp. 54-59.

BERNERS-LEE, T. (2000) *Building the Future* [online]. World Wide Web Consortium. Available from: http://www.w3.org/2000/Talks/0906-xmlweb-tbl/slide9-0.html. [Accessed 22 January 2005].

BERNERS-LEE, T. (2006) *Linked Data - Design Issues* [online]. Available from: http://www.w3.org/DesignIssues/LinkedData.html. [Accessed 20 October 2009].

BERNERS-LEE, T., HENDLER, J. & LASSILA, O. (2001) The Semantic Web. *Scientific American,* **284**(5), pp. 34-43.

BERNSTEIN, P., BRODIE, M., CERI, S., DEWITT, FRANKLIN, M., GARCIA-MOLINA, H., GRAY, J., HELD, J., HELLERSTEIN, J., JAGADISH, H. V., LESK, M., MAIER, D., NAUGHTON, J., PIRAHESH, H., STONEBRAKER, M. & ULLMAN, J. (1998) The Asilomar Report on Database Research. *ACM SIGMOD Record,* **27**(4), pp. 74-80.

BERRUETA, D., LABRA, J. E. & POLO, L. (2006) Searching over Public Administration Legal Documents Using Ontologies. In: *Proceedings of Seventh Joint Conference on Knowledge-Based Software Engineering. 2006*. pp. 167-175.

BERTINO, E., CATANIA, B. & ZARRI, G. P. (2001) The latest developments - Ch. 4. *Intelligent Database Systems.* 1st ed. Harlow, UK, Addison-Wesley Professional, pp. 369-403.

BHAGDEV, R., CHAPMAN, S., CIRAVEGNA, F., LANFRANCHI, V. & PETRELLI, D. (2008) Hybrid Search: Effectively Combining Keywords and Semantic Searches. In: *Proceedings of 5th European Semantic Web Conference – ESWC 2008. Tenerife, Spain, 1-5 June 2008*. pp. 554-568.

BHATTACHERJEE, A. (1998) Management of Emerging Technologies: Experiences and Lessons Learned at US WEST. *Information & Management,* **33**(5), pp. 263-272.

BHOGAL, J., MACFARLANE, A. & SMITH, P. (2007) A review of ontology based query expansion. *Information Processing and Management,* **43**(4), pp. 866-886.

BIZER, C. (2009) The Emerging Web of Linked Data. *IEEE Intelligent Systems,* **24**(5), pp. 87-92.

BORST, W. N. (1997) *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. Ph.D. Thesis, SIKS - Dutch Graduate School for Information and Knowledge Systems.

BOUQUET, P., GIUNCHIGLIA, F., VAN HARMELEN, F., SERAFINI, L. & STUCKENSCHMIDT, H. (2003) C-OWL: Contextualizing Ontologies. In: *Proceedings of 2nd International Semantic Web Conference (ISWC 2003). Sanibel Island, Florida, USA, October 2003*. Springer Verlag, Lecture Notes in Computer Science 2870, pp. 164-179.

BRACHMAN, R. J. (1983) What IS-A Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks. *IEEE Computer: special issue on knowledge representation,* **16**(10), pp. 30-36.

BRIGHT, M. W. (1994) Automated Resolution of Semantic Heterogeneity in Multidatabases. *ACM Transactions on Database Systems (TODS),* **19**(2), pp. 212-253.

BRODIE, M. L. (2002) The Grand Challenge of Information Technology. In: *Proceedings of an Invited Talk, OntoWeb Meeting. Innsbruck, Austria, December 16-18 2002*. Verizon Information Technology.

BRODIE, M. L. (2003) *Enterprise Level Integration - It's Not About Technology* [online]. Verizon Information Technology. Available from: http://www.plmdc.engin.umich.edu/BrodieIntegration.pdf. [Accessed 29 December 2004].

BUKHRES, O., ELMAGARMID, A., GHERFAL, F. F., LIU, X., BARKER, K. & SCHALLER, T. (1996) The Integration of Database Systems. In BUKHRES, O. & ELMAGARMID, A. (eds.) *Object-Oriented Multidatabase Systems.* Prentice-Hall, Englewood Cliffs, NJ, pp. 37-56.

CHEN, P. P. S. (1976) The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS),* **1**(1), pp. 9-36.

CLEVERDON, C. W. (1991) The signifiance of the Cranfield tests on index languages. In: *Proceedings of 14th annual international ACM SIGIR conference on Research and development in information retrieval. Chicago, Illinois, United States, 1991*. pp. 3-12.

CONNOLLY, D., VAN HARMELEN, F., HORROCKS, I., MCGUINNESS, D. L., PATEL-SCHNEIDER, P. F. & STEIN, L. A. (2001) *DAML+OIL (March 2001) Reference Description* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/daml+oil-reference. [Accessed 18 January 2005].

CYCORP (2005) *OpenCyc – A General Knowledge Base* [online]. Available from: http://www.opencyc.org/. [Accessed 22 August 2005].

DE BRUIJN, J. (2003) *Using Ontologies - Enabling Knowledge Sharing and Reuse on the Semantic Web* [online]. Innsbruck, Austria, DERI - Digital Enterprise Research Institute. Available from: http://www.deri.ie/publications/techpapers/documents/DERI-TR-2003-10-29.pdf. [Accessed 15 October 2007].

DE MICHELIS, G., DUBOIS, E., JARKE, M., MATTHES, F., MYLOPOULOS, J., PAPAZOGLOU, M., POHL, K., SCHMIDT, J., WOO, C. & YU, E. (1997) *Cooperative Information Systems: A Manifesto* [online]. In PAPAZOGLOU, M. P. & SCHLAGETER, G. (Eds.) Cooperative Information System: Trends and Directions. Academic Press, pp. 315-363. Available from: http://citeseer.ist.psu.edu/michelis97cooperative.html. [Accessed 13 July 2005].

DECKER, S., MELNIK, S., VAN HARMELEN, F., FENSEL, D., KLEIN, M., BROEKSTRA, J., ERDMANN, M. & HORROCKS, I. (2000a) The Semantic Web: The roles of XML and RDF. *IEEE Internet Computing,* **15**(3), pp. 63-74.

DECKER, S., MITRA, P. & MELNIK, S. (2000b) Framework for the Semantic Web: An RDF Tutorial. *IEEE Internet Computing,* **4**(6), pp. 68-73.

DING, L., FININ, T., JOSHI, A., PAN, R., COST, R. S., PENG, Y., REDDIVARI, P., DOSHI, V. & SACHS, J. (2004) Swoogle: A Search and Metadata Engine for the Semantic Web. In: *Proceedings of ACM Thirteenth Conference on Information and Knowledge Management (CIKM04). Washington, DC, November 2004*. ACM, pp. 652-659.

DING, L., FININ, T., JOSHI, A., PENG, Y., PAN, R. & REDDIVARI, P. (2005) Search on the Semantic Web. *IEEE Computer,* **38**(10), pp. 62-69.

DING, Y. & FOO, S. (2002) Ontology Research and Development: Part 2 - A Review of Ontology mapping and evolving. *Journal of Information Science,* **28**(5), pp. 383-396.

DMOZ (2008) *Open Directory Project* [online]. Netscape Communication Corporation. Available from: http://www.dmoz.org/. [Accessed 26 September 2008].

DREW, P., KING, R., MCLEOD, D., RUSINKIEWICZ, M. & SILBERSCHATZ, A. (1993) Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems. *ACM SIGMOD Record,* **22**(3), pp. 47-56.

ELIOT, S. & BARLOW, J. (2002) KM Can Enable the Enterprise. In: *Proceedings of KMWorld & Intranets Conference. Santa Clara Convention Center, Santa Clara, California, 31 October 2002*. IBM Lotus Software Group.

ELMASRI, R. & NAVATHE, S. B. (2004) *Fundamentals of Database Systems*. 4th ed., Boston, MA: Pearson Addison-Wesley, 0-321-20448-4.

FANG, W.-D., ZHANG, L., WANG, Y.-X. & DONG, S.-B. (2005) Toward a semantic search engine based on ontologies. In: *Proceedings of Fourth International Conference on Machine Learning and Cybernetics. Guanzhou, China, 18-21 August, 2005 2005*. pp. 1913-1918.

FELLBAUM, C. (1998) *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press, 0-262-06197-X.

FENSEL, D., HORROCKS, I., VAN HARMELEN, F., MCGUINNESS, D. L. & PATEL-SCHNEIDER, P. F. (2001) OIL: An Ontology Infrastructure for the Semantic Web. *IEEE Intelligent Systems,* **16**(2), pp. 38-45.

FIKES, R. & KEHLER, T. (1985) The Role of Frame-Based Representation in Reasoning. *Communications of the ACM,* **28**(9), pp. 904-920.

GARCIA-MOLINA, H., PAPAKONSTANTINOU, Y., QUASS, D., RAJARAMAN, A., SAGIV, Y., ULLMAN, J. D., VASSALOS, V. & WIDOM, J. (1997) The TSIMMIS approach to mediation: data models and languages. *Journal of Intelligent Information Systems,* **8**(2), pp. 117-132.

GARCIA-SOLACO, M., SALTOR, F. & CASTELLANOS, M. (1996) Semantic heterogeneity in multidatabase systems. In BUKHRES, O. & ELMAGARMID, A. (eds.) *Object-Oriented Multidatabase Systems.* Prentice-Hall, Englewood Cliffs, NJ, pp. 129-202.

GILBERT, J. & BUTLER, M. H. (2003) *Review of existing tools for working with schemas, metadata, and thesauri* [online]. Palo Alto, CA., Hewlett-Packard Labs. Available from: http://www.hpl.hp.com/techreports/2003/HPL-2003-218.pdf. [Accessed 23 November 2004].

GLIGOROV, R., TEN KATE, W., ALEKSOVSKI, Z. & VAN HARMELEN, F. (2007) Using Google distance to weight approximate ontology matches. In: *Proceedings of 16th international conference on World Wide Web. Banff, Alberta, Canada, 8-12 May 2007*. pp. 767-776.

GÓMEZ-PÉREZ, A. & CORCHO, O. (2002) Ontology Languages for the Semantic Web. *IEEE Intelligent Systems,* **17**(1), pp. 54-60.

GONZALO, J., VERDEJO, F., CHUGUR, I. & CIGARRÁN, J. (1998) Indexing with WordNet synsets can improve text retrieval. In: *Proceedings of Coling-ACL 1998 Workshop on Usage of WordNet in Natural Language Processing Systems. Monteal, Canada, 1998*. pp. 38-44.

GOOGLE (2008) *Corporate Information* [online]. Available from: http://www.google.com/corporate/tech.html. [Accessed 28 August 2008].

GRAU, B. C., PARSIA, B. & SIRIN, E. (2006) Combining OWL ontologies using *E*-Connections. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web,* **4**(1), pp. 40-59.

GRAY, P., ATKINSON, M., GOBLE, C., KAY, M., KERRIDGE, J., MOODY, K. & KING, P. (2000) *CPHC Workshop on Research Directions for RAE Themes* [online]. Manchester, UK. Available from: http://www.csd.abdn.ac.uk/~pgray/man.html. [Accessed 3 October 2004].

GRUBER, T. R. (1992) *Ontolingua: A Mechanism to Support Portable Ontologies* [online]. Knowledge Systems Laboratory, Stanford University. Available from: http://www.ksl.stanford.edu/KSL_Abstracts/KSL-91-66.html. [Accessed 17 June 2005].

GRUBER, T. R. (1993) A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition,* **5**(2), pp. 199-220.

GRUBER, T. R. (1995) Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies,* **43**(5-6), pp. 907-928.

GUARINO, N. (1998) Formal Ontology and Information Systems. In: *Proceedings of 1st International Conference on Formal Ontologies in Information Systems (FOIS'98). Trento, Italy, 6-8 June 1998*. IOS Press, pp. 3-15.

GUDIVADA, V. N., RAGHAVAN, V. V., GROSKY, W. I. & KASANAGOTTU, R. (1997) Information Retrieval on the World Wide Web. *IEEE Internet Computing,* **1**(5), pp. 58-68.

HAKIA (2008) *Search Engine Beta* [online]. Available from: http://www.hakia.com/. [Accessed 3 March 2008].

HAMMER, J. & MCLEOD, D. (1993) An Approach to Resolving Semantic Heterogeneity in a Federation of Autonomous, Heterogeneous Database Systems. *International Journal of Intelligent & Cooperative Information Systems,* **2**(1), pp. 51-83.

HART, G., DOLBEAR, C., GOODWIN, J. & KOVACS, K. (2007) *Domain Ontology Development* Southampton, Ordnance Survey Research. Available from: http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/publications/docs/2007/Domain_Ontology_Development_V1.pdf. [Accessed 22 October 2007].

HAWKING, D., VOORHEES, E., CRASWELL, N. & BAILEY, P. (2000) *Overview of the TREC-8 Web Track* [online]. National Institute of Standards and Technology. Available from: http://trec.nist.gov/pubs/trec8/papers/web_overview.pdf. [Accessed 4 April 2008].

HENDLER, J., BERNERS-LEE, T. & MILLER, E. (2002) Integrating Applications on the Semantic Web. *Journal of the Institute of Electrical Engineers of Japan,* **122**(10), pp. 676-680.

HENDLER, J. & MCGUINNESS, D. L. (2000) The DARPA Agent Markup Language. *IEEE Intelligent Systems,* **15**(6), pp. 67-73.

HORRIDGE, M., KNUBLAUCH, H., RECTOR, A., STEVENS, R. & WROE, C. (2004) *A Practical Guide to Building OWL Ontologies Using The Protégé-Owl Plugin and Co-ODE Tools Edition 1.0* [online]. University of Manchester. Available from: http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf. [Accessed 2 November 2004].

HORROCKS, I. (2002) DAML+OIL: a Description Logic for the SemanticWeb. *IEEE Data Engineering Bulletin,* **25**(1), pp. 4-9.

HORROCKS, I., FENSEL, D., BROEKSTRA, J., DECKER, S., ERDMANN, M., GOBLE, C., HARMELEN, F. V., KLEIN, M., STAAB, S., STUDER, R. & MOTTA, E. (2000) *The Ontology Inference Layer OIL* [online]. Available from: http://www.cs.man.ac.uk/~horrocks/Publications/download/2000/oil.pdf. [Accessed 14 March 2005].

HP-LABS (2005) *HP Labs Semantic Web Research* [online]. Hewlett Packhard Labs. Available from: http://www.hpl.hp.com/semweb/. [Accessed January 21 2005].

JOHNSON MCMANUS, D. & SNYDER, C. A. (2003) Synergy between data warehousing and knowledge management: three industries reviewed. *Int. J. Information Technology and Management,* **2**(1/2), pp. 85-99.

KALINICHENKO, L., MISSIKOFF, M., SCHIAPPELLI, F. & SKVORTSOV, N. (2003) *Ontological Modeling* [online]. St. Petersburg, Russia, 5th Russian Conference on Digital Libraries (RCDL2003). Available from: http://rcdl2003.spbu.ru/proceedings/D2.pdf. [Accessed 21 January 2005].

KARLSBJERG, J. & DAMSGAARD, J. (2001) Make or buy - A Taxonomy of Intranet Implementation Strategies. In: *Proceedings of 9th European Conference on Information Systems. Bled, Slovenia, June 27-29 2001*. Dept. of Computer Science, Aalborg Univ., pp. 579-592.

KASHYAP, V., SHAH, K. & SHETH, A. (1995) *Metadata for building the MultiMedia Patch Quilt* [online]. Available from: http://citeseer.ist.psu.edu/cache/papers/cs/2020/http:zSzzSzra.cs.uga.eduzSz~amitzSz67-PatchQuilt.pdf/kashyap95metadata.pdf. [Accessed 1 June 2005].

KASHYAP, V. & SHETH, A. (1996) Semantic and schematic similarities between database objects: a context-based approach. *The VLDB Journal,* **5**(4), pp. 276-304.

KASHYAP, V. & SHETH, A. (2000) *Information Brokering across Heterogeneous Digital Data: A Metadata-based Approach*. Boston, MA: Kluwer Academic Publishers Group, 0792378830.

KIM, H. H. (2005) ONTOWEB: Implementing an ontology-based Web retrieval system. *Journal of the American Society for Information Science and Technology,* **56**(11), pp. 1167-1176.

KIM, W. (1991) Object-oriented database systems: strengths and weaknesses. *Journal of Object-Oriented Programming,* **1**(4), pp. 21-29.

KIM, W., CHOI, I., GALA, S. & SCHEEVEL, M. (1993) On Resolving Schematic Heterogeneity in Multidatabase Systems. *Distributed and Parallel Databases,* **1**(3), pp. 251-279.

KLUSCH, M. (2001) Information agent technology for the Internet: A survey. *Data & Knowledge Engineering,* **36**(3), pp. 337-372.

KNOBLOCK, C. A., ARENS, Y. & HSU, C.-N. (1994) Cooperating Agents for Information Retrieval. In: *Proceedings of Second International Conference on Cooperative Information Systems. Toronto, Ontario, Canada, 1994*. University of Toronto Press, pp. 122-133.

KNUBLAUCH, H. (2003) *An AI tool for the real world - Knowledge modeling with Protégé* [online]. JavaWorld. Available from: http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege_p.html. [Accessed 23 December 2004].

LAMB, R. & DAVIDSON, E. (2000) The New Computing Archipelago: Intranet Islands of Practice. In: *Proceedings of IFIP Working Group 8.2 Conference - The Social and Organizational Perspective on Research and Practice in Information Technology. Aalborg, Denmark, 10-12 June 2000*. Kluwer Academic Publishers, pp. 255-274.

LASSILA, O. & MCGUINNESS, D. (2001) *The Role of Frame-Based Representation on the Semantic Web* [online]. Technical Report KSL-01-02, Knowledge Systems Laboratory, Stanford University, CA. Available from: http://www.ep.liu.se/ea/cis/2001/005/cis01005.pdf. [Accessed 12 July 2005].

LEI, Y., UREN, V. & MOTTA, E. (2006) SemSearch: A Search Engine for the Semantic Web. In: *Proceedings of 15th International Conference - Managing Knowledge in a World of Networks - EKAW 2006. Podebrady, Czech Republic, 2-6 October 2006*. Springer Berlin / Heidelberg, pp. 238-245.

LENAT, D. B. (1995) CYC: A Large-Scale Investment in Knowledge Infrastructure. *Communications of the ACM,* **38**(11), pp. 32-38.

LEVY, A. Y. (1999) Logic-Based Techniques In Data Integration. In: *Proceedings of Workshop on Logic-Based Artificial Intelligence (LBAI). Washington, DC, June 14-16 1999*. Computer Science Department, University of Maryland.

LEVY, A. Y., RAJARAMAN, A. & ORDILLE, J. J. (1996) Querying Heterogeneous Information Sources Using Source Descriptions. In: *Proceedings of 22nd International Conference on Very Large Databases (VLDB). Mumbai (Bombay), India, 1996*. VLDB Endowment, Saratoga, CA, pp. 251-262.

LORENZ, B., OHLBACH, H. J. & YANG, L. (2005) Ontology of Transportation Networks. *REWERSE - Reasoning on the Web with Rules and Semantics.* Department of Computer Science, University of Munich.

MCBRIDE, B. (2002) Jena: A Semantic Web Toolkit. *IEEE Internet Computing,* **6**(6), pp. 55-59.

MCCOOL, R. (2005) Rethinking the semantic Web. Part 1. *IEEE Internet Computing,* **9**(6), pp. 88, 86-87.

MCCOOL, R. (2006) Rethinking the semantic Web, Part 2. *IEEE Internet Computing,* **10**(1), pp. 96, 93-95.

MCILRAITH, S., SON, T. C. & ZENG, H. (2001) Semantic Web Services. *IEEE Intelligent Systems,* **16**(2), pp. 46-53.

MIKA, P. (2008) Microsearch: An Interface for Semantic Search. In: *Proceedings of Workshop on Semantic Search (SemSearch 2008) at ESWC 2008: 5th European Semantic Web Conference. Tenerife, Spain, 2 June 2008*. pp. 79-88.

NARDI, D. & BRACHMAN, R. J. (2002) An Introduction to Description Logics. In BAADER, F., CALVANESE, D., MCGUINNESS, D. L., NARDI, D. & PATEL-SCHNEIDER, P. F. (eds.) *The Description Logic Handbook.* Cambridge University Press, pp. 5-44.

NAVIGLI, R. & VELARDI, P. (2003) An analysis of ontology-based query expansion strategies. In: *Proceedings of International Workshop on Adaptive Text Extraction and Mining at 14th European Conference on Machine Learning and 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. Dubrovnik, Croatia, 22 September 2003*. pp. 42-49.

NILES, I. & PEASE, A. (2001) Towards a Standard Upper Ontology. In: *Proceedings of 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001). Ogunquit, Maine, October 17-19 2001*. ACM Press, New York, pp. 2-9.

NOY, N. F. & HAFNER, C. D. (1997) The State of the Art in Ontology Design - A Survey and Comparative Review. *AI Magazine,* **18**(3), pp. 53-74.

NOY, N. F. & KLEIN, M. (2002) *Ontology Evolution: Not the Same as Schema Evolution* [online]. Appeared in Knowledge and Information Systems 2004, Vol. 6(4) pp. 428-440. Available from: http://www.smi.stanford.edu/pubs/SMI_Reports/SMI-2002-0926.pdf. [Accessed 29 January 2005].

NOY, N. F., SINTEK, M., DECKER, S., CRUZBEZY, M., FERGERSON, R. W. & MUSEN, M. A. (2001) Creating Semantic Web Contents with Protégé-2000. *IEEE Intelligent Systems,* **16**(2), pp. 60-71.

ONTOWEB (2002) Evaluation of Ontology-based Tools. In: *Proceedings of OntoWeb-SIG3 Workshop (EON2002), 13th International Conference on Knowledge Engineering and Knowledge Management In Cooperation with AAAI, EKAW02. Siguenza, Spain, 30 September 2002*. CEUR-WS, pp. 1-139.

OREN, E., DELBRU, R., CATASTA, M., CYGANIAK, R., STENZHORN, H. & TUMMARELLO, G. (2008) *Sindice.com: A Document-oriented Lookup Index for Open Linked Data* [online]. Available from: http://www.eyaloren.org/pubs/ijmso2008.pdf. [Accessed 7 October 2008].

OUKSEL, A. & SHETH, A. (1999) Semantic Interoperability in Global Information Systems. *ACM SIGMOD Record,* **28**(1), pp. 5-12.

PARENT, C. & SPACCAPIETRA, S. (2000) *Database Integration: The Key to Data Interoperability* [online]. in Advances in Object-Oriented Data Modeling, M. P. Papazoglou, S. Spaccapietra, Z. Tari (Eds.), The MIT Press, 2000. Available from: http://lbdsun.epfl.ch/e/publications_new/articles.pdf/OObook.pdf. [Accessed 21 February 2005].

PATEL-SCHNEIDER, P. & FENSEL, D. (2002) Layering the Semantic Web: Problems and Directions. In: *Proceedings of 1st International Semantic Web Conference (ISWC 2002). Sardinia, Italy, June 2002*.

POWELL, T. A. (2002) *Web Design: The Complete Reference, 2nd Edition*. McGraw-Hill/Osbourne, 0-07-222422-8.

POWERSET (2008) *Powerlabs* [online]. Available from: http://www.powerset.com/. [Accessed 7 May 2008].

RECTOR, A. L. (2003) Modularisation of domain ontologies implemented in description logics and related formalisms including OWL. In: *Proceedings of 2nd International Conference On Knowledge Capture. Sanibel Island, FL, USA, 2003*. ACM Press, New York, NY, USA, pp. 121-128.

ROB, P. & CORONEL, C. (2002) *Database Systems Design, Implementation and Management*. Fifth Edition ed.: Course Technology, 0-619-06269-X.

ROCHA, C., SCHWABE, D. & DE ARAGÃO, M. P. (2004) A Hybrid Approach for Searching in the Semantic Web. In: *Proceedings of 13th international conference on World Wide Web. New York, New York, USA, 17-22 May 2004*. pp. 374-383.

ROTH, M. A., WOLFSON, D. C., KLEEWEIN, J. C. & NELIN, C. J. (2002) Information integration: A new generation of information technology. *IBM Systems Journal,* **41**(4), pp. 563-577.

SALTON, G., WONG, A. & YANG, J. (1975) A vector space model for automatic indexing. *Communications of the ACM,* **18**(11), pp. 613-620.

SHETH, A. P. (1998) Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In GOODCHILD, M. F., EGENHOFER, M. J., FEGEAS, R. & KOTTMAN, C. A. (eds.) *Interoperating Geographic Information Systems.* Dordrecht, Netherlands, Kluwer, Academic Publishers, pp. 5-30.

SHETH, A. P. & LARSON, J. A. (1990) Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys (CSUR) - Special issue on heterogeneous databases,* **22**(3), pp. 183-236.

SIGIR (2008) *International Conference on Research and Development in Information Retrieval (SIGIR)* [online]. Available from: http://www.informatik.uni-trier.de/~ley/db/conf/sigir/. [Accessed 4 April 2008].

SINGHAL, A. (2001) Modern Information Retrieval: A Brief Overview. *IEEE Computer Society Data Engineering Bulletin,* **24**(4), pp. 35-43.

SMEATON, J. (2002) *IBM's Own Intranet: Saving Big Blue Millions* [online]. Intranet Journal. Available from: http://www.intranetjournal.com/articles/200209/pij_09_25_02a.html. [Accessed 8 January 2005].

SPÄRCK JONES, K. (2004) A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation,* **60**(5), pp. 493-502.

SPARCK JONES, K., WALKER, S. & ROBERTSON, S. E. (2000) A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management,* **36**(6), pp. 779-808.

SPYNS, P., MEERSMAN, R. & JARRAR, M. (2002) Data modelling versus Ontology engineering. *ACM SIGMOD Record - Special Section on Semantic Web and Data Management,* **31**(4), pp. 12-17.

STEVENS, R., GOBLE, C. A. & BECHHOFER, S. (2000) Ontology-based knowledge representation for bioinformatics. *Briefings in Bioinformatics,* **1**(4), pp. 398-414.

STONEBRAKER, M., AGRAWAL, R., DAYAL, U., NEUHOLD, E. J. & REUTER, A. (1993) DBMS Research at a Crossroads: The Vienna Update. In: *Proceedings of 19th International Conference on Very Large Databases (VLDB). Dublin, Ireland, 24-27 August 1993*. pp. 688-692.

STUCKENSCHMIDT, H. & KLEIN, M. (2004) Structure-Based Partitioning of Large Concept Hierarchies. In: *Proceedings of Third International Semantic Web Conference – ISWC 2004. Hiroshima, Japan, 2004*. Lecture Notes in Computer Science, pp. 289-303.

STUDER, R., BENJAMINS, V. R. & D.FENSEL (1998) Knowledge Engineering: Principles and Methods. *Data & Knowledge Engineering,* **25**(1-2), pp. 161-197.

SWEO (2009) *Linking Open Data* [online]. W3C-SWEO. Available from: http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData. [Accessed 20 October 2009].

SYCARA, K., PAOLUCCI, M., SOUDRY, J. & SRINIVASAN, N. (2004) Dynamic Discovery and Coordination of Agent-Based Semantic Web Services. *IEEE Internet Computing,* **8**(3), pp. 66-73.

TIUN, S., ABDULLAH, R. & KONG, T. E. (2001) Automatic Topic Identification Using Ontology Hierarchy. In: *Proceedings of Second International Conference on Computational Linguistics and Intelligent Text Processing. Mexico-City, Mexico, 18-24 February 2001*. pp. 444-453.

TREC (2008) *Text REtrieval Conference* [online]. Available from: http://trec.nist.gov/. [Accessed 4 April 2008].

UNSPSC (1998) *The United Nations Standard Products and Services Code* [online]. Available from: http://www.unspsc.org/Defaults.asp. [Accessed 5 September 2005].

USCHOLD, M. & GRUNINGER, M. (2004) Ontologies and Semantics for Seamless Connectivity. *ACM SIGMOD Record - Special Issue on Semantic Integration,* **33**(4), pp. 58-64.

VAN RIJSBERGEN, C. J. (1979) *Information Retrieval* [online]. London, Butterworths. Available from: http://www.dcs.gla.ac.uk/Keith/Preface.html. [Accessed 20 April 2008].

VOORHEES, E. (1994) Query expansion using lexical-semantic relations. In: *Proceedings of 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 1994*. pp. 61-69.

W3C (2002) *RDF Vocabulary Description Language 1.0: RDF Schema* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/2002/WD-rdf-schema-20021112/. [Accessed 10 May 2006].

W3C (2004a) *Architecture of the World Wide Web, Volume One* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/2004/REC-webarch-20041215/. [Accessed 4 January 2004].

W3C (2004b) *OWL Web Ontology Language Guide* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/owl-guide/. [Accessed 24 October 2007].

W3C (2004c) *RDF Primer* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/rdf-primer/. [Accessed 15 November 2007].

W3C (2004d) *RDF Vocabulary Description Language 1.0: RDF Schema* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/. [Accessed 24 January 2006].

W3C (2005) *SPARQL Query Language for RDF* [online]. World Wide Web Consortium. Available from: http://www.w3.org/TR/rdf-sparql-query/. [Accessed 24 January 2006].

W3C (2008) *Semantic Web Case Studies and Use Cases* [online]. World Wide Web Consortium. Available from: http://www.w3.org/2001/sw/sweo/public/UseCases/. [Accessed 24 March 2008].

WACHE, H., VÖGELE, T., VISSER, U., STUCKENSCHMIDT, H., SCHUSTER, G., NEUMANN, H. & HÜBNER, S. (2001) Ontology-Based Integration of Information — A Survey of Existing Approaches. In: *Proceedings of IJCAI-01 Workshop on Ontologies and Information Sharing. Seattle, WA., 2001*. pp. 108-117.

WAGNER, W. P., CHUNG, Q. B. & BARATZ, T. (2002) Implementing corporate intranets: lessons learned from two high-tech firms. *Industrial Management & Data Systems*, **102**(3), pp. 140-145.

WIEDERHOLD, G. (1992) Mediators in the Architecture of Future Information Systems. *IEEE Computer,* **25**(3), pp. 38-49.

WIEDERHOLD, G. (1999) Mediation to Deal with Heterogeneous Data Sources. In: *Proceedings of Interoperating Geographic Information Systems: Second International Workshop (Interop'99). Zurich, Switzerland, March 10-12 1999*. Springer-Verlag Telos, pp. 1-16.

BIBLIOGRAPHY

ANTONIOU, G. & VAN HARMELEN, F. (2004) A Semantic Web Primer. Cambridge, Massachusetts: MIT Press, ISBN: 0-262-01210-3.

POWERS, S. (2003) Practical RDF. Sebastopol, CA: O'Reilly & Associates, Inc., ISBN: 0-596-00263-7.

APPENDICES

APPENDIX A: GLOSSARY

ABox. The Description Logics ABox construct is an “assertional construct” that specifies assertions on individuals, i.e. ABox statements are associated with instances of TBox defined classes, e.g. class City hasCity ∋ “London”. ABox constructs are thus TBox-compliant statements about a vocabulary - see also TBox.

Antonym. A word opposite in meaning to another, e.g. large and small, wide and narrow.

Axiom. A statement or truth, i.e. an axiom in an OWL ontology is a sentence in First-Order Logic that is assumed to be true without proof.

Black-box testing. Unlike white-box testing (see glossary item), black-box testing requires no knowledge of an application’s internal program structure; instead, it involves development of test case data inputs, based on the function of the program, to validate outputs against pre-determined outcomes.

Context (Ontology)**.** For the purposes of this research, a context is a modular, self-standing, topic specific or query context-relevant small ontology.

Cyc. Developed by Cycorp, the Cyc Knowledge Server is considered the largest and most complete, multi-contextual knowledge base and inference engine - http://www.cyc.com/cyc.

DAML. A U.S. government-sponsored project (DARPA Agent Markup Language), created as an extension to XML and RDF and an early de facto standard ontology language that provided greater description expressiveness than basic RDF Schema.

DAML+OIL. An acronym for DAML plus Ontology Inference Layer: a development of the DAML ontology language incorporating the OIL layer.

DARPA. The U.S. government-sponsored Defence Advanced Research Programs Agency responsible for the development of new technology for use by the military - http://www.darpa.mil/.

DC. An acronym for The Dublin Core Metadata Initiative: related to bibliographic and digital information and provides a set of basic metadata properties (e.g. title, creator, rights) and demonstrates a vocabulary for classifying Web resources - http://purl.org/DC/.

DB. An abbreviation for database and can refer to any form of data record keeping e.g. text, and spreadsheet. Often used to refer to database management system (DBMS).

DBMS. Software (database management system) to manage and retrieve data in a database. Provides transparency between physical data and application programs, i.e. DB users and other programs are not required to understand where the data is physically located and, in a multi-user system, which other users may be accessing the data.

Design Autonomy. Where a designer creates a database model from a conceptual model of a real world situation. Different designers will influence designs, which inevitably results in systems heterogeneity.

DL. An acronym for Description Logic, a sub-language of predicate logic. A knowledge representation language for formally representing domain terminology by construction of complex concept and role descriptions. DL permits both representation of domain facts, through *assertions,* and the derivation of new facts through rules of *inference*.

DMOZ. An acronym for Directory Mozilla, also known as The Open Directory Project: a comprehensive human-reviewed classification directory of the web – see http://dmoz.org/. DMOZ drives directory services for many key information portals and search engines.

Domain. Used in the context of an Ontology domain, e.g. medicine, geography, and engineering. Also used to specify the *domain* of a property, e.g. hasRole *Transportation*.

DQL. The DAML Query Language provided a language and protocol for "agent-to-agent query-answering", i.e. query agent (client) and answering agent (server); using knowledge represented in DAML+OIL ontologies. With the development of OWL, DQL has been superseded by OWL-QL.

E-R. The Entity-Relationship database model is a conceptual data model that views the real world as entities having attributes and relationships. A basic component of the model is the E-R diagram, employed to visually represent data objects. E-R model constructs are subsequently transformed into relational tables that are normalised to minimise redundancy.

FaCT. A Description Logic ontology-reasoning and classification tool (Fast Classification of Terminologies), released under GNU public license. It can be used in conjunction with Protégé OWL - http://www.cs.man.ac.uk/~horrocks/FaCT/.

Federated DB. A multidatabase system, i.e. a collection or federation of heterogeneous and geographically disparate database systems that operate autonomously locally, but which may export various elements of data schemas for sharing and use by members of the federation. Primarily focused on systems, structure and semantic interoperability and providing a balance between shared data integration and federated user autonomy.

Foundation (Ontology). A foundation (upper) ontology, serving as a starting point, or "anchor", for developing and mapping concepts and relationships to domain and application ontologies. Various examples include Cyc, SUMO, DOLCE.

Frame. A frame (e.g. Frame-based Ontology) is a structured, named data object (class), or set of objects used to represent some concept in a domain; having a set of slots (object attributes or properties), some of which may be pointers to other frames.

GIS. A term used for Geographic Information Systems (geo-spatial), although often used to separately mean a Global Information System (world-wide or universal).

Heterogeneity. In information systems, heterogeneity refers to data from disparate sources that represent a similar real world context, whilst demonstrating syntactic, structural, or semantic conflicts - in systems, language, conceptual modelling and schema design approach.

Homonym. This term describes where the same name is used for unrelated or semantically different entities - as opposed to a different name for an equivalent entity (synonym). For example, a *table* of data and *table* furniture. See also hypernym, and hyponym.

Hypernym. A word or term that defines a super ordinate or super class, e.g. animal is a hypernym (broader term or generalisation) of tiger. Hypernym is the opposite of hyponym - see also homonym and synonym.

Hyponym. The opposite of hypernym. A term defining a sub class (narrower term or specialisation), e.g. geographic is a narrower term for spatial. See homonym and synonym.

Information space. An abstract concept representing everything accessible via the Web.

Internet. A global network of networks through which computers communicate, by cables or wireless links, by sending information in packets.

io. Represents "instance of", i.e. an individual or instance of a class, e.g. "London" is an instance or member of the class Capital. Instances are individual objects of classes that define types of objects.

is-a or isa. Used to describe a domain and denote relationships in class hierarchies (and implicit inheritance). The term has origins in early Semantic Networks research. An RDF Schema triple example would be (class) Motorway (is a) subClassOf Highway.

JADE. The Java Agent DEvelopment Framework is an open-source software framework to facilitate the development of Multi-Agent Systems (MAS) - http://jade.tilab.com/.

JDBC. Java Database Connectivity is the industry standard for database-independent connectivity between the Java language and a range of databases. JDBC is used to establish a connection, send SQL statements, and process results with a database.

Jena. An open-source Java application-programming interface (API) for Semantic Web applications, developed by HP Labs. The Jena toolkit uses packages that provide Java libraries for developer use in a programmatic environment based on RDF, RDF Schema, DAML+OIL, and OWL technologies. Jena2 facilitates persistence (storage) of RDF and OWL models (through the use of back-end database engines) and a reasoning sub-system.

KIF. The Knowledge Interchange Format is a computer-oriented language for the interchange of knowledge among disparate programs. It has *declarative* semantics (i.e. the meaning of expressions in the representation can be understood without requiring interpretation. http://www.ksl.stanford.edu/knowledge-sharing/kif/.

KR Ontology. The Knowledge Representation Ontology (upper ontology) demonstrates basic categories that have been derived from a variety of sources in logic, linguistics, philosophy, and artificial intelligence - see http://www.jfsowa.com/ontology/toplevel.htm.

MDA. This refers to the language and platform independent Model Driven Architecture (MDA), a core application-modelling standard from the OMG (Object Management Group). This software-engineering tool features the Unified Modelling Language (UML) and has its origins in object-oriented (OO) modelling.

MDBS. Multi-database systems, being either *homogeneous* systems: containing a single logical database that is physically distributed and managed by a single distributed database management system, or *heterogeneous* systems: containing diverse systems, models and languages, including legacy systems.

Meronym. A meronym denotes a semantic relation that describes a part of a whole, or a member of something, e.g. “wing” is a meronym of “aeroplane”.

Metadata. A pivotal component in the Semantic Web, where information is described by metadata annotations; using ontology vocabularies that specify concepts (classes) and their roles (relations), to give meaning to data. Metadata summarises information content to provide a metadata context. Metadata makes use of ontologies and represents the abstraction of data content. See also Metadata and Resource Description at W3C - http://www.w3.org/Metadata/.

Modularity. A characteristic demonstrated in ontology design, where large ontologies may be sub-divided into specific domains. This may then result in the need to consider the task of integration of component ontologies for specific purposes. Therefore, modularity is important in potentially large ontologies, to facilitate re-use, interchange, evolution and maintenance.

MySQL. An open-source database management system that also includes the facility to store RDF as N-triples.

Namespace. An XML namespace is created to provide a unique identifier (namespaces help to avoid tag "collisions"), by a URI reference, i.e. the Web address of a resource. A namespace is declared using reserved attributes (either "xmlns" or "xmlns:"). See also "Namespaces in XML 1.0" - http://www.w3.org/TR/REC-xml-names/.

Namespace Prefix. Every resource namespace (URI) can be conveniently represented in short form by declaring a prefix that is bound to a full URI path, e.g. a prefix dg is created by the declaration xmlns:dg="http://url_address". This reduces code and makes namespace changes more manageable.

OIL. The Ontology Inference Layer: offered greater expressivity by including precise semantics for describing term meanings and thus also for describing implied information through inference. OIL was later combined with DAML to produce the richer DAML+OIL.

O-O. In this report, object-oriented relates to O-O databases, where information is represented by classes and class objects, their properties, and inheritance (of super class attributes and methods) and encapsulation (the ability of an object to hide its data and methods from the rest of the world).

Ontology. In computing, ontologies represent a formal vocabulary for capturing domain knowledge (i.e. the universe of discourse at whatever level) by specifying concepts, their attributes, relationships between concepts, and constraints on relationships. As Ontology represents a domain theory for information sharing, it must therefore be a shared and consensual vocabulary.

OntoViz. An ontology visualisation tool that can be included in Protégé - http://protege.cim3.net/cgi-bin/wiki.pl?OntoViz.

OpenCyc. The open source version of the Cyc general knowledge base (KB) and reasoning engine. The KB browser URL is: http://opencyc1.cyc.com:3602/cgi-bin/cyccgi/cg?cb-start. Interestingly, an OWL version of OpenCyc takes about 9 hours to load into Protégé.

OQE. A method of (ontology-based) query expansion, or query augmentation, to add additional query relevant terms to the initial query, using a query context relevant taxonomy or ontology. Query expansion can improve retrieval results by addressing the problems of word ambiguity in natural language and the use of single terms to convey the context of an information source required.

OS. Ordnance Survey: the national mapping agency of Great Britain. A key activity is "Semantic Reference Systems", i.e. combining multiple data sources so they can be exploited in new ways - http://www.ordnancesurvey.co.uk/oswebsite/partnerships/research/.

OWA. An Ontology functions on the principle of Open World Assumptions (i.e. something cannot be assumed to be false unless proved to be so). Whereas, a database operates on closed world assumptions (e.g. assumes that everything not known is false, or anything not found does not exist).

OWL. The W3Cs 2004 Web Ontology Language recommendation for the Semantic Web, where information is given explicit meaning; making it easier for machines to automatically process and integrate Web information, instead of simply presenting information to humans. OWL offers three incrementally expressive species: OWL Lite, OWL DL (Description Logic) and OWL FULL.

OWL-QL. Is a W3C candidate standard formal language for deductive query answering of OWL-based ontologies on the Semantic Web. OWL-QL precisely specifies the semantic relationships between a query, a query answer, and the ontology. Unlike standard structured query languages, OWL-QL supports query-answering dialogues in which an answering agent may use automated reasoning for answers, i.e. it facilitates inferencing capability to derive new data from data already known. OWL-QL is an updated version of DQL.

OWLViz. The CO-ODE group designed the OWLViz OWL visualisation tool to be used as a Protégé OWL plug-in; it produces a graphical representation of class hierarchies - http://www.co-ode.org/downloads/owlviz/.

Prolog. A programming language centred on pattern matching, tree-based structures, and reasoning; well suited to problems that involve objects and relations – available in various implementations, e.g. SWI-Prolog - http://www.swi-prolog.org/.

PrologTab. An integration of GNU Prolog for Java with Protégé-2000, where Protégé relations are represented as facts within Prolog - http://prologtab.sourceforge.net/.

Protégé. A Java-based, open-source knowledge base and ontology development tool/editor that has evolved from projects conducted at the Medical Informatics group at Stanford University. Available as free software under the open-source Mozilla Public License.

RACER. A Description Logic ontology reasoning (classification and inference) system for use with OWL. Unlike FaCT, RACER is essentially a commercial product.

RDF. A general-purpose, declarative language (Resource Description Framework) for representing information in the Web. It provides a standard approach for using the universal XML syntax to represent metadata in the form of statements about properties and relationships of items on the Web. RDF defines the meaning of data, rather than simply providing data containers, and provides a framework specification for constructing logical languages that can work together in the Semantic Web, e.g. RDF Schema (a simple RDF ontology vocabulary modelling language) and OWL.

RDQL. A query language for RDF and RDF Schema that has been superseded by W3C's recommendation SPARQL.

Reasoner. See FaCT and RACER reasoner/classifiers.

Resource. Anything to which an identity can be attached via a URI, e.g. a Web page or page element, an image, an RDF file and component objects and properties.

RuleML. The Rule Markup Language is part of the Rule Markup Initiative to define shared rules in XML for deduction, rewriting, and further inferential-transformational tasks. Rules are used for various purposes, including: engineering diagnosis, commercial business rules, and legal reasoning. See - http://www.ruleml.org/.

Schema. A structural description of the type of facts held in a database. The schema describes the entities represented in a database, their attributes, constraints and relationships.

SDLC. An acronym for System Development Life-cycle, and the development of information systems through a recognised process of feasibility and requirements investigation, analysis and design, testing, implementation and maintenance.

Semantic conflict. Semantic conflicts occur whenever two information or data repositories do not use the same interpretation of common information; possibly as a result of differing structural representations of concepts, or differing solutions resulting in naming conflicts, e.g. synonyms, homonyms, hypernyms, and hyponyms.

Semantic Web. Referred to as the next generation of the Web: to create a universal medium, a Web of shared data and information that is underpinned with descriptions, or meaning, so that data can be shared and processed by intelligent machines or Web agents, as well as by humans; to ultimately provide an automated knowledge resource that accurately reflects real world meaning or semantics.

SPARQL. A recursive acronym for the SPARQL Protocol And RDF Query Language; a W3C recommendation query language that supersedes RDQL. SPARQL is a client-server based RDF query language that functions by matching RDF graph patterns and permits disjunction in the query; allowing more complex query processing than RDQL. It provides no inference capability, i.e. to derive new data from data already known.

Spatial. Relates to the general concept of space: in terms of distribution, distance, direction, areas and other aspects of space on the Earth's surface. Whilst often used in the context of geo-spatial and associated with geographic information systems, spatial has broader, encompassing meaning than geography (the study of the surface of the earth).

SQL. A formal, structured language to retrieve data from a relational database (Structured Query Language). Similarly, object relational databases (O-R) databases can be interrogated using Object Query Language (OQL).

SUMO. A foundation ontology (Suggested Upper Merged Ontology) developed by the IEEE Standard Upper Ontology Working Group used for research and applications in search, linguistics and reasoning. Whilst SUMO and its domain ontologies represent a smaller, more abstract theory of all things than Cyc, SUMO has been mapped to the entire WordNet lexicon.

SWRL. An acronym for the Semantic Web Rule Language: that combines OWL and RuleML - http://www.w3.org/Submission/SWRL/. SWRL is useful because it adds rules to OWL DL that provide more expressive power over Description Logic.

Synonym. A term to describe the use of a different name for an equivalent entity; as opposed to the same name for different entities (homonym). For example, an *airline* and a *carrier*. See also homonym, hypernym, and hyponym.

Tag. A reference to an XML tag, where the tag is bound to data e.g. "customer" as in <customer fname="Sid" />.

TBox. The Description Logics TBox (terminological) construct: defines a domain in terms of a controlled vocabulary, describing assertions on concepts and class hierarchies, e.g. $\mathrm{LargeSettlement} = \forall(\mathrm{City} \sqcup \mathrm{Town})$. TBox constructs - see also ABox.

Triple. An RDF triple is a statement consisting of a subject (resource object O), a predicate (the subject's attribute A), and an object (or value V), e.g. Child hasParent Mother. It is also termed as a binary relation A(O,V). RDF triples form a node-arc-node structure. An alternative form of representation is possible, using N-triples (line-based, plain text format); these are suitable for storage in databases, e.g. MySQL, Oracle.

UML. Is an acronym for Unified Modelling Language: a platform-independent application-modelling standard that can also be used to model ontologies.

UNSPSC. An acronym for the United Nations Standard Products and Services Code: a formal taxonomy of products and services; it displays strict class "inheritance".

URI. A Uniform Resource Identifier (a short string also referred to as UR*L* - *locator*) to identify, or name, resources like documents, images, files, or services. As the URI often starts with http://, it can provide a unique identifying reference that also serves as a URL; assuming the resource is physically present at the address represented by the URI. See also "Naming and Addressing: URIs, URLs" - http://www.w3.org/Addressing/.

W3C. An acronym for the World Wide Web Consortium: an international consortium where member organisations collaborate to lead the World Wide Web to its full potential, by developing Web protocol standards and guidelines.

Web Services. Defined by the W3C as a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks, over a network.

White-box testing. Known also as glass-box testing, white-box testing refers to software program development tesing, using the internal programming structures and algorithms.

WordNet. A structured collection of English language terms, developed by the Cognitive Science Laboratory at Princeton University, and forming an online lexical reference system, where nouns, verbs, adjectives and adverbs are organised into synonym sets.

WWW. Known also as "Web" or "W3", the World Wide Web started as an information project at CERN and has gradually developed into a global resource of network-accessible information relating to human knowledge. The Web is traversed by using hypertext and communication protocols.

XML. Extensible Markup Language is W3C's generic language for creating new markup languages: to represent data in a nested, treelike structure. XML tags are not predefined and therefore rely on users defining their own tags. XML is accepted as the de facto standard for data exchange on the Web; particularly in business-to-business data transfers (b2b), and forms the universal syntax upon which RDF is constructed. See - http://www.w3.org/XML/.

APPENDIX B: ONTOLOGY QUERY EXPANSION ALGORITHMS

This section contains the following Java-based ontology query expansion algorithms developed using the Jena Ontology API.

1. Inheritance Class Hierarchy Algorithm – pages XI to XVII.

   Fig. A1. *S+S OQE* Part 1.

   Fig. A2. *S+S OQE* Part 2.

   Fig. A3. *S+S OQE* Part 3.

   Fig. A4. *S+S OQE* Part 4.

   Fig. A5. *S+S OQE* Part 5.

   Fig. A6. *S+S OQE* Part 6.

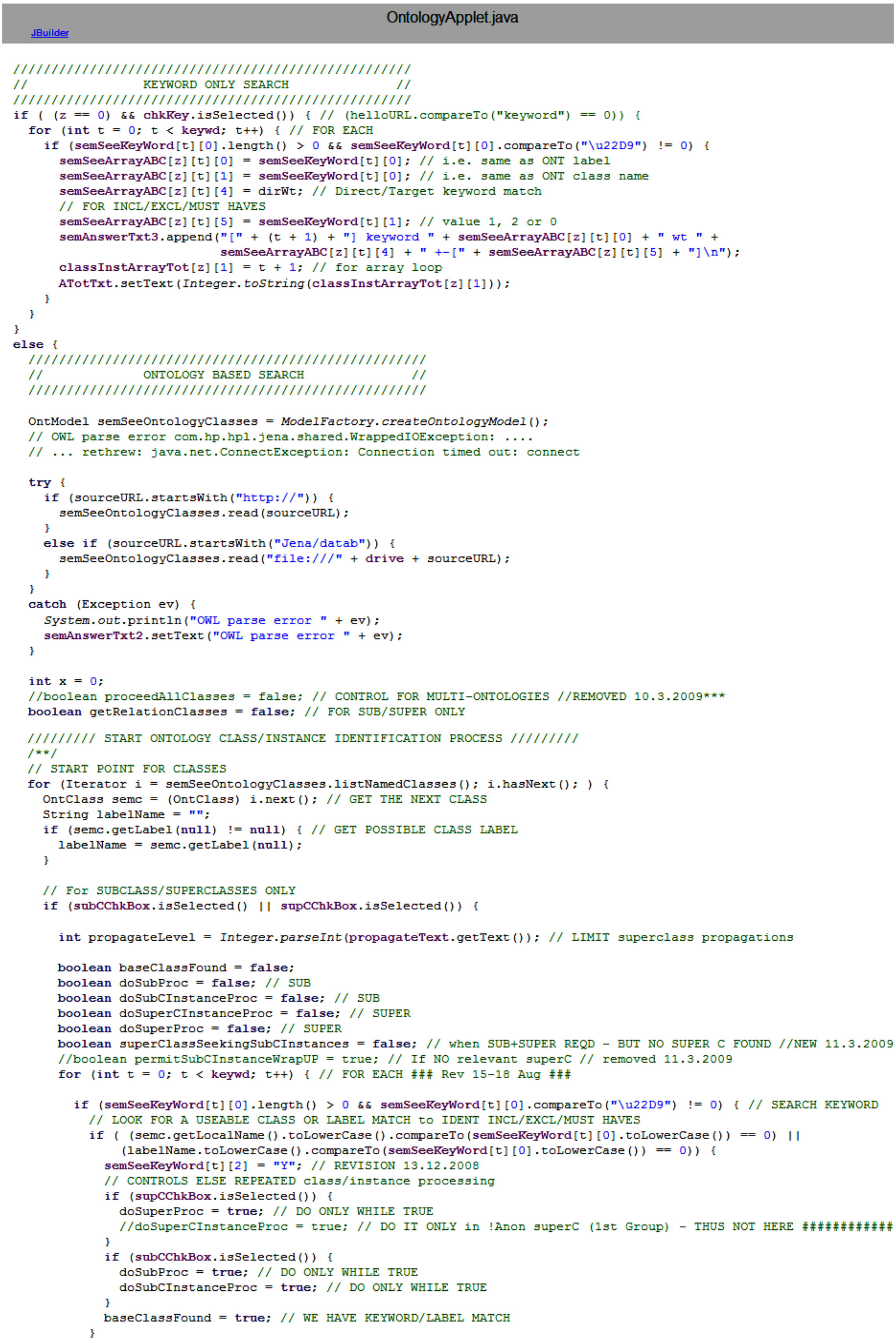
   Fig. A7. *S+S OQE* Part 7.

2. Relation Class Algorithm stage 1 and 2 – pages XVIII and XIX.

   Fig. A8. *S+S+R OQE* stage 1.

   Fig. A9. *S+S+R OQE* stage 2.

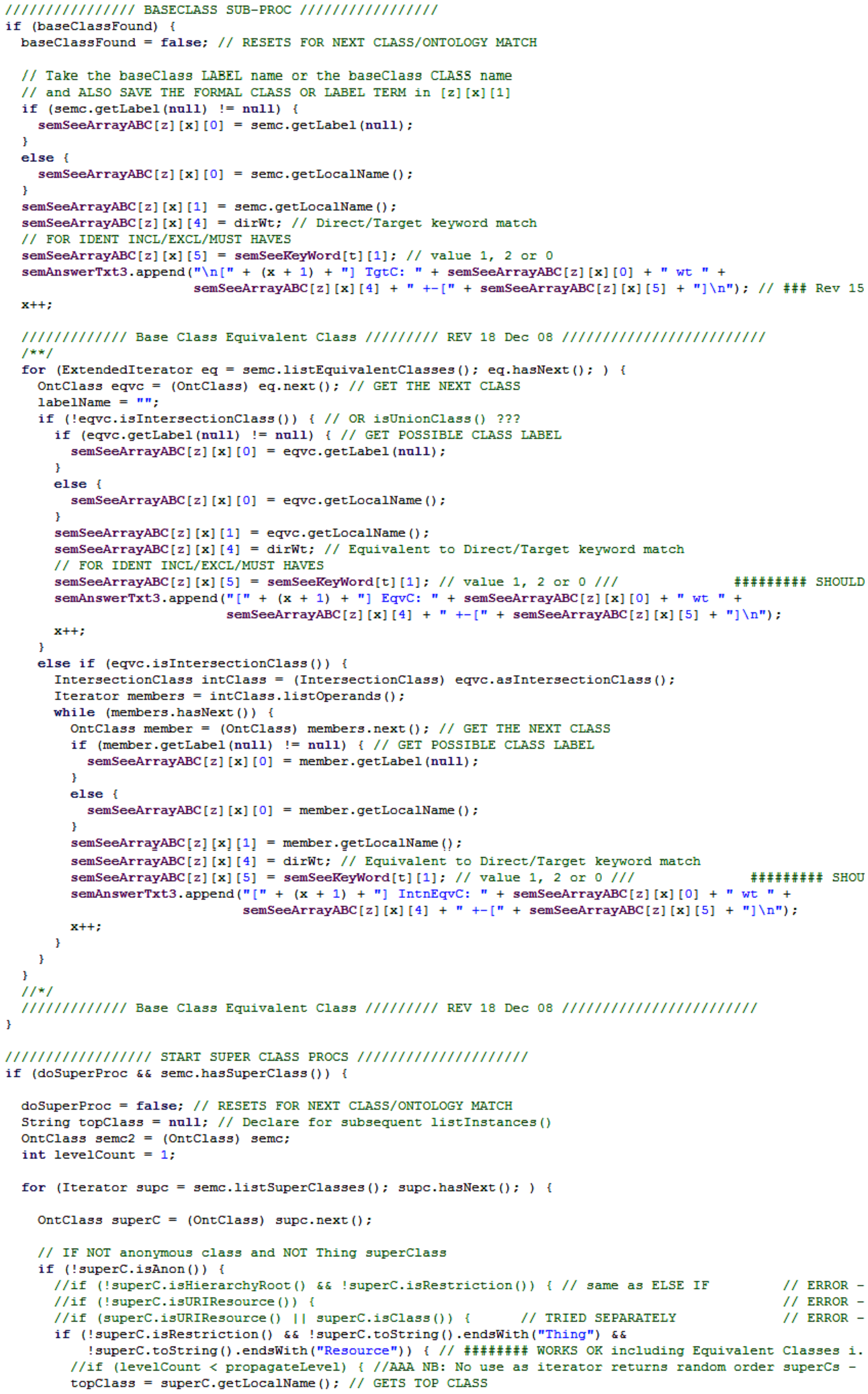
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes).



```
////////////////////////////////////////////////////
//               KEYWORD ONLY SEARCH              //
////////////////////////////////////////////////////
if ( (z == 0) && chkKey.isSelected()) { // (helloURL.compareTo("keyword") == 0)) {
  for (int t = 0; t < keywd; t++) { // FOR EACH
    if (semSeeKeyWord[t][0].length() > 0 && semSeeKeyWord[t][0].compareTo("\u22D9") != 0) {
      semSeeArrayABC[z][t][0] = semSeeKeyWord[t][0]; // i.e. same as ONT label
      semSeeArrayABC[z][t][1] = semSeeKeyWord[t][0]; // i.e. same as ONT class name
      semSeeArrayABC[z][t][4] = dirWt; // Direct/Target keyword match
      // FOR INCL/EXCL/MUST HAVES
      semSeeArrayABC[z][t][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
      semAnswerTxt3.append("[" + (t + 1) + "] keyword " + semSeeArrayABC[z][t][0] + " wt " +
                           semSeeArrayABC[z][t][4] + " +-[" + semSeeArrayABC[z][t][5] + "]\n");
      classInstArrayTot[z][1] = t + 1; // for array loop
      ATotTxt.setText(Integer.toString(classInstArrayTot[z][1]));
    }
  }
}
else {
  ////////////////////////////////////////////////////
  //             ONTOLOGY BASED SEARCH              //
  ////////////////////////////////////////////////////

  OntModel semSeeOntologyClasses = ModelFactory.createOntologyModel();
  // OWL parse error com.hp.hpl.jena.shared.WrappedIOException: ....
  // ... rethrew: java.net.ConnectException: Connection timed out: connect

  try {
    if (sourceURL.startsWith("http://")) {
      semSeeOntologyClasses.read(sourceURL);
    }
    else if (sourceURL.startsWith("Jena/datab")) {
      semSeeOntologyClasses.read("file:///" + drive + sourceURL);
    }
  }
  catch (Exception ev) {
    System.out.println("OWL parse error " + ev);
    semAnswerTxt2.setText("OWL parse error " + ev);
  }

  int x = 0;
  //boolean proceedAllClasses = false; // CONTROL FOR MULTI-ONTOLOGIES //REMOVED 10.3.2009***
  boolean getRelationClasses = false; // FOR SUB/SUPER ONLY

  ///////// START ONTOLOGY CLASS/INSTANCE IDENTIFICATION PROCESS /////////
  /**/
  // START POINT FOR CLASSES
  for (Iterator i = semSeeOntologyClasses.listNamedClasses(); i.hasNext(); ) {
    OntClass semc = (OntClass) i.next(); // GET THE NEXT CLASS
    String labelName = "";
    if (semc.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
      labelName = semc.getLabel(null);
    }

    // For SUBCLASS/SUPERCLASSES ONLY
    if (subCChkBox.isSelected() || supCChkBox.isSelected()) {

      int propagateLevel = Integer.parseInt(propagateText.getText()); // LIMIT superclass propagations

      boolean baseClassFound = false;
      boolean doSubProc = false; // SUB
      boolean doSubCInstanceProc = false; // SUB
      boolean doSuperCInstanceProc = false; // SUPER
      boolean doSuperProc = false; // SUPER
      boolean superClassSeekingSubCInstances = false; // when SUB+SUPER REQD - BUT NO SUPER C FOUND //NEW 11.3.2009
      //boolean permitSubCInstanceWrapUP = true; // If NO relevant superC // removed 11.3.2009
      for (int t = 0; t < keywd; t++) { // FOR EACH ### Rev 15-18 Aug ###

        if (semSeeKeyWord[t][0].length() > 0 && semSeeKeyWord[t][0].compareTo("\u22D9") != 0) { // SEARCH KEYWORD
          // LOOK FOR A USEABLE CLASS OR LABEL MATCH to IDENT INCL/EXCL/MUST HAVES
          if ( (semc.getLocalName().toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0) ||
              (labelName.toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0)) {
            semSeeKeyWord[t][2] = "Y"; // REVISION 13.12.2008
            // CONTROLS ELSE REPEATED class/instance processing
            if (supCChkBox.isSelected()) {
              doSuperProc = true; // DO ONLY WHILE TRUE
              //doSuperCInstanceProc = true; // DO IT ONLY in !Anon superC (1st Group) - THUS NOT HERE ############
            }
            if (subCChkBox.isSelected()) {
              doSubProc = true; // DO ONLY WHILE TRUE
              doSubCInstanceProc = true; // DO ONLY WHILE TRUE
            }
            baseClassFound = true; // WE HAVE KEYWORD/LABEL MATCH
          }
```

Fig. A1. *S+S OQE* Part 1.

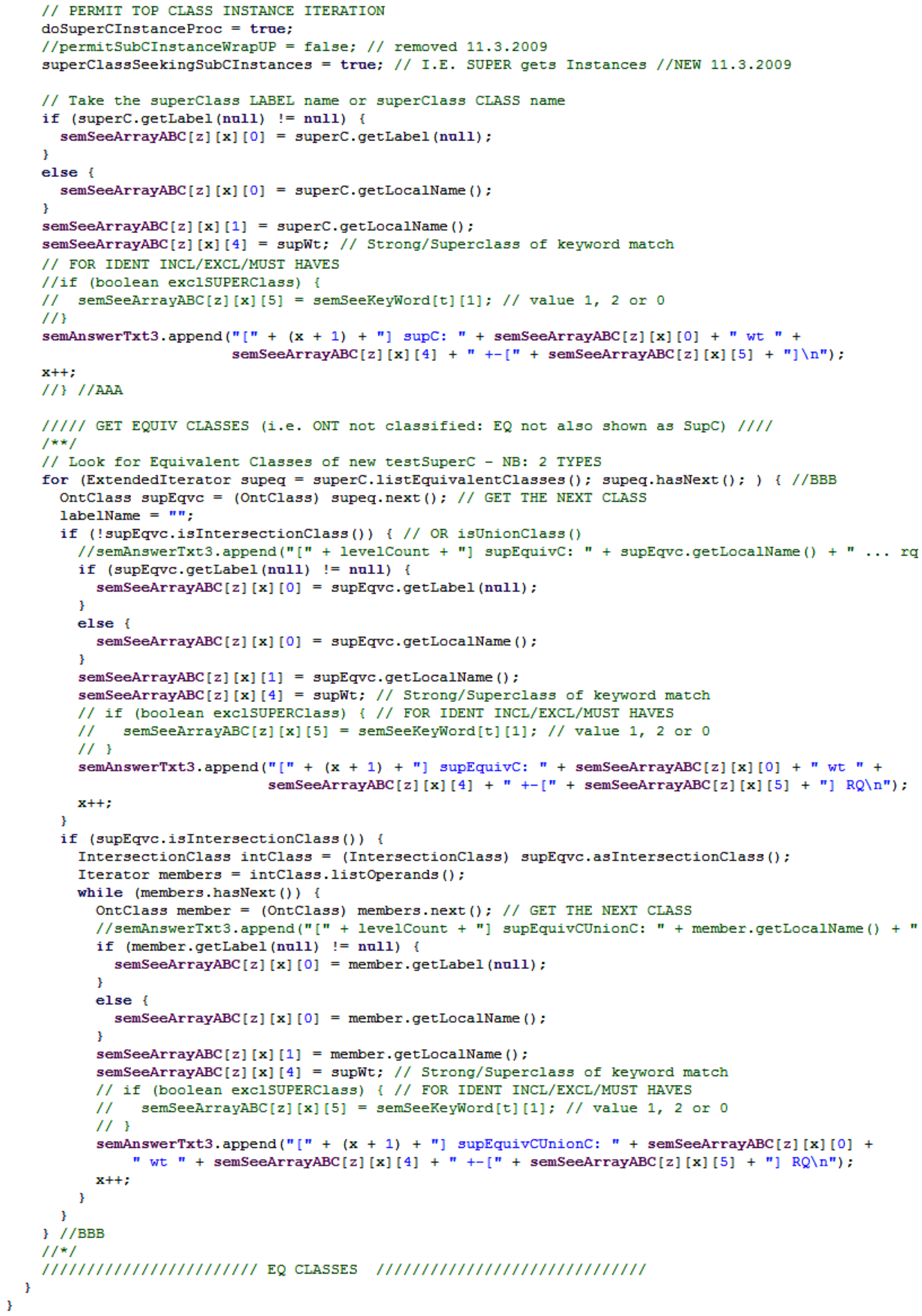
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
//////////////// BASECLASS SUB-PROC /////////////////
if (baseClassFound) {
  baseClassFound = false; // RESETS FOR NEXT CLASS/ONTOLOGY MATCH

  // Take the baseClass LABEL name or the baseClass CLASS name
  // and ALSO SAVE THE FORMAL CLASS OR LABEL TERM in [z][x][1]
  if (semc.getLabel(null) != null) {
    semSeeArrayABC[z][x][0] = semc.getLabel(null);
  }
  else {
    semSeeArrayABC[z][x][0] = semc.getLocalName();
  }
  semSeeArrayABC[z][x][1] = semc.getLocalName();
  semSeeArrayABC[z][x][4] = dirWt; // Direct/Target keyword match
  // FOR IDENT INCL/EXCL/MUST HAVES
  semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
  semAnswerTxt3.append("\n[" + (x + 1) + "] TgtC: " + semSeeArrayABC[z][x][0] + " wt " +
                       semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n"); // ### Rev 15
  x++;

  ///////////// Base Class Equivalent Class ///////// REV 18 Dec 08 /////////////////////////
  /**/
  for (ExtendedIterator eq = semc.listEquivalentClasses(); eq.hasNext(); ) {
    OntClass eqvc = (OntClass) eq.next(); // GET THE NEXT CLASS
    labelName = "";
    if (!eqvc.isIntersectionClass()) { // OR isUnionClass() ???
      if (eqvc.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
        semSeeArrayABC[z][x][0] = eqvc.getLabel(null);
      }
      else {
        semSeeArrayABC[z][x][0] = eqvc.getLocalName();
      }
      semSeeArrayABC[z][x][1] = eqvc.getLocalName();
      semSeeArrayABC[z][x][4] = dirWt; // Equivalent to Direct/Target keyword match
      // FOR IDENT INCL/EXCL/MUST HAVES
      semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0 ///             ######### SHOULD
      semAnswerTxt3.append("[" + (x + 1) + "] EqvC: " + semSeeArrayABC[z][x][0] + " wt " +
                           semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
      x++;
    }
    else if (eqvc.isIntersectionClass()) {
      IntersectionClass intClass = (IntersectionClass) eqvc.asIntersectionClass();
      Iterator members = intClass.listOperands();
      while (members.hasNext()) {
        OntClass member = (OntClass) members.next(); // GET THE NEXT CLASS
        if (member.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
          semSeeArrayABC[z][x][0] = member.getLabel(null);
        }
        else {
          semSeeArrayABC[z][x][0] = member.getLocalName();
        }
        semSeeArrayABC[z][x][1] = member.getLocalName();
        semSeeArrayABC[z][x][4] = dirWt; // Equivalent to Direct/Target keyword match
        semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0 ///             ######### SHOU
        semAnswerTxt3.append("[" + (x + 1) + "] IntnEqvC: " + semSeeArrayABC[z][x][0] + " wt " +
                             semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
        x++;
      }
    }
  }
  //*/
  ///////////// Base Class Equivalent Class ///////// REV 18 Dec 08 ////////////////////////
}

////////////////// START SUPER CLASS PROCS /////////////////////
if (doSuperProc && semc.hasSuperClass()) {

  doSuperProc = false; // RESETS FOR NEXT CLASS/ONTOLOGY MATCH
  String topClass = null; // Declare for subsequent listInstances()
  OntClass semc2 = (OntClass) semc;
  int levelCount = 1;

  for (Iterator supc = semc.listSuperClasses(); supc.hasNext(); ) {

    OntClass superC = (OntClass) supc.next();

    // IF NOT anonymous class and NOT Thing superClass
    if (!superC.isAnon()) {
      //if (!superC.isHierarchyRoot() && !superC.isRestriction()) { // same as ELSE IF         // ERROR -
      //if (!superC.isURIResource()) {                                                        // ERROR -
      //if (superC.isURIResource() || superC.isClass()) {      // TRIED SEPARATELY            // ERROR -
      if (!superC.isRestriction() && !superC.toString().endsWith("Thing") &&
          !superC.toString().endsWith("Resource")) { // ######## WORKS OK including Equivalent Classes i.
        //if (levelCount < propagateLevel) { //AAA NB: No use as iterator returns random order superCs -
        topClass = superC.getLocalName(); // GETS TOP CLASS
```

Fig. A2. *S+S OQE* Part 2.

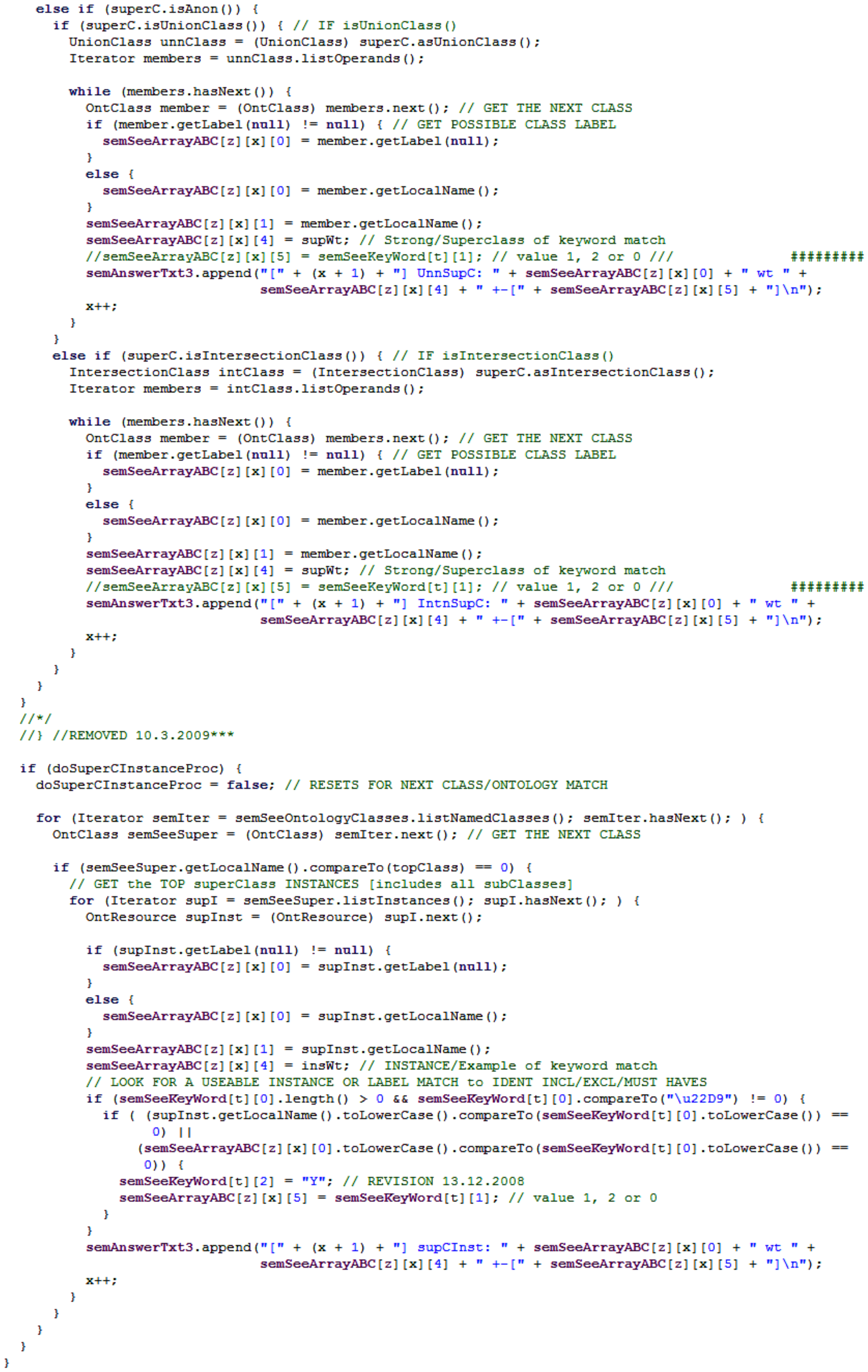
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
    // PERMIT TOP CLASS INSTANCE ITERATION
    doSuperCInstanceProc = true;
    //permitSubCInstanceWrapUP = false; // removed 11.3.2009
    superClassSeekingSubCInstances = true; // I.E. SUPER gets Instances //NEW 11.3.2009

    // Take the superClass LABEL name or superClass CLASS name
    if (superC.getLabel(null) != null) {
      semSeeArrayABC[z][x][0] = superC.getLabel(null);
    }
    else {
      semSeeArrayABC[z][x][0] = superC.getLocalName();
    }
    semSeeArrayABC[z][x][1] = superC.getLocalName();
    semSeeArrayABC[z][x][4] = supWt; // Strong/Superclass of keyword match
    // FOR IDENT INCL/EXCL/MUST HAVES
    //if (boolean exclSUPERClass) {
    //  semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
    //}
    semAnswerTxt3.append("[" + (x + 1) + "] supC: " + semSeeArrayABC[z][x][0] + " wt " +
                         semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
    x++;
    //} //AAA

    ///// GET EQUIV CLASSES (i.e. ONT not classified: EQ not also shown as SupC) ////
    /**/
    // Look for Equivalent Classes of new testSuperC - NB: 2 TYPES
    for (ExtendedIterator supeq = superC.listEquivalentClasses(); supeq.hasNext(); ) { //BBB
      OntClass supEqvc = (OntClass) supeq.next(); // GET THE NEXT CLASS
      labelName = "";
      if (!supEqvc.isIntersectionClass()) { // OR isUnionClass()
        //semAnswerTxt3.append("[" + levelCount + "] supEquivC: " + supEqvc.getLocalName() + " ... rq
        if (supEqvc.getLabel(null) != null) {
          semSeeArrayABC[z][x][0] = supEqvc.getLabel(null);
        }
        else {
          semSeeArrayABC[z][x][0] = supEqvc.getLocalName();
        }
        semSeeArrayABC[z][x][1] = supEqvc.getLocalName();
        semSeeArrayABC[z][x][4] = supWt; // Strong/Superclass of keyword match
        // if (boolean exclSUPERClass) { // FOR IDENT INCL/EXCL/MUST HAVES
        //   semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
        // }
        semAnswerTxt3.append("[" + (x + 1) + "] supEquivC: " + semSeeArrayABC[z][x][0] + " wt " +
                             semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "] RQ\n");
        x++;
      }
      if (supEqvc.isIntersectionClass()) {
        IntersectionClass intClass = (IntersectionClass) supEqvc.asIntersectionClass();
        Iterator members = intClass.listOperands();
        while (members.hasNext()) {
          OntClass member = (OntClass) members.next(); // GET THE NEXT CLASS
          //semAnswerTxt3.append("[" + levelCount + "] supEquivCUnionC: " + member.getLocalName() + "
          if (member.getLabel(null) != null) {
            semSeeArrayABC[z][x][0] = member.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = member.getLocalName();
          }
          semSeeArrayABC[z][x][1] = member.getLocalName();
          semSeeArrayABC[z][x][4] = supWt; // Strong/Superclass of keyword match
          // if (boolean exclSUPERClass) { // FOR IDENT INCL/EXCL/MUST HAVES
          //   semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
          // }
          semAnswerTxt3.append("[" + (x + 1) + "] supEquivCUnionC: " + semSeeArrayABC[z][x][0] +
              " wt " + semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "] RQ\n");
          x++;
        }
      }
    } //BBB
    //*/
    //////////////////////// EQ CLASSES  //////////////////////////////
  }
}
```

Fig. A3. *S+S OQE* Part 3.

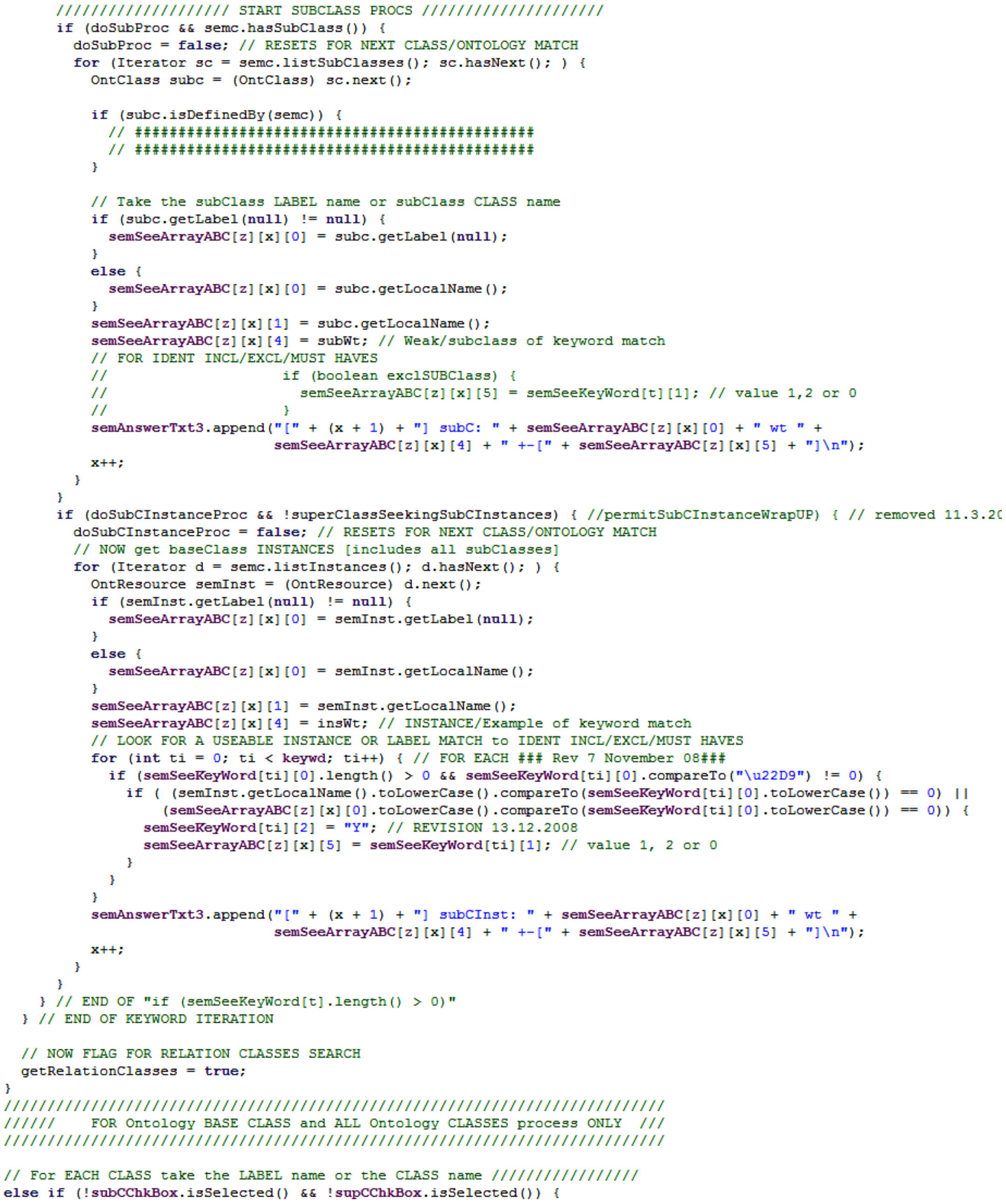
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
    else if (superC.isAnon()) {
      if (superC.isUnionClass()) { // IF isUnionClass()
        UnionClass unnClass = (UnionClass) superC.asUnionClass();
        Iterator members = unnClass.listOperands();

        while (members.hasNext()) {
          OntClass member = (OntClass) members.next(); // GET THE NEXT CLASS
          if (member.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
            semSeeArrayABC[z][x][0] = member.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = member.getLocalName();
          }
          semSeeArrayABC[z][x][1] = member.getLocalName();
          semSeeArrayABC[z][x][4] = supWt; // Strong/Superclass of keyword match
          //semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0 ///              #########
          semAnswerTxt3.append("[" + (x + 1) + "] UnnSupC: " + semSeeArrayABC[z][x][0] + " wt " +
                               semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
          x++;
        }
      }
      else if (superC.isIntersectionClass()) { // IF isIntersectionClass()
        IntersectionClass intClass = (IntersectionClass) superC.asIntersectionClass();
        Iterator members = intClass.listOperands();

        while (members.hasNext()) {
          OntClass member = (OntClass) members.next(); // GET THE NEXT CLASS
          if (member.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
            semSeeArrayABC[z][x][0] = member.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = member.getLocalName();
          }
          semSeeArrayABC[z][x][1] = member.getLocalName();
          semSeeArrayABC[z][x][4] = supWt; // Strong/Superclass of keyword match
          //semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0 ///              #########
          semAnswerTxt3.append("[" + (x + 1) + "] IntnSupC: " + semSeeArrayABC[z][x][0] + " wt " +
                               semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
          x++;
        }
      }
    }
  }
  //*/
  //} //REMOVED 10.3.2009***

  if (doSuperCInstanceProc) {
    doSuperCInstanceProc = false; // RESETS FOR NEXT CLASS/ONTOLOGY MATCH

    for (Iterator semIter = semSeeOntologyClasses.listNamedClasses(); semIter.hasNext(); ) {
      OntClass semSeeSuper = (OntClass) semIter.next(); // GET THE NEXT CLASS

      if (semSeeSuper.getLocalName().compareTo(topClass) == 0) {
        // GET the TOP superClass INSTANCES [includes all subClasses]
        for (Iterator supI = semSeeSuper.listInstances(); supI.hasNext(); ) {
          OntResource supInst = (OntResource) supI.next();

          if (supInst.getLabel(null) != null) {
            semSeeArrayABC[z][x][0] = supInst.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = supInst.getLocalName();
          }
          semSeeArrayABC[z][x][1] = supInst.getLocalName();
          semSeeArrayABC[z][x][4] = insWt; // INSTANCE/Example of keyword match
          // LOOK FOR A USEABLE INSTANCE OR LABEL MATCH to IDENT INCL/EXCL/MUST HAVES
          if (semSeeKeyWord[t][0].length() > 0 && semSeeKeyWord[t][0].compareTo("\u22D9") != 0) {
            if ( (supInst.getLocalName().toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) ==
                  0) ||
                (semSeeArrayABC[z][x][0].toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) ==
                 0)) {
              semSeeKeyWord[t][2] = "Y"; // REVISION 13.12.2008
              semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
            }
          }
          semAnswerTxt3.append("[" + (x + 1) + "] supCInst: " + semSeeArrayABC[z][x][0] + " wt " +
                               semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
          x++;
        }
      }
    }
  }
}
```

Fig. A4. *S+S OQE* Part 4.

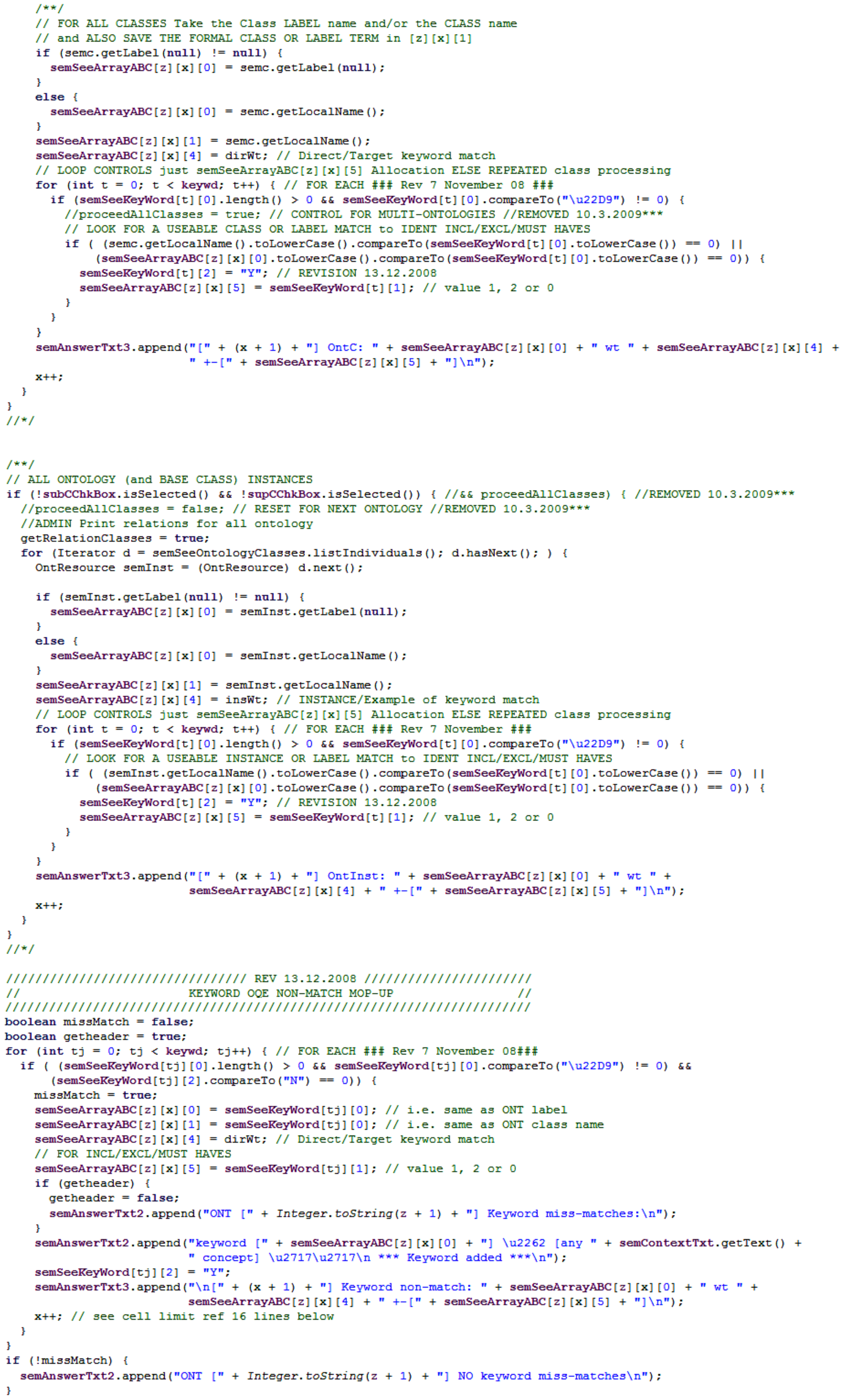
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
      ///////////////////// START SUBCLASS PROCS /////////////////////
      if (doSubProc && semc.hasSubClass()) {
        doSubProc = false; // RESETS FOR NEXT CLASS/ONTOLOGY MATCH
        for (Iterator sc = semc.listSubClasses(); sc.hasNext(); ) {
          OntClass subc = (OntClass) sc.next();

          if (subc.isDefinedBy(semc)) {
            // ##############################################
            // ##############################################
          }

          // Take the subClass LABEL name or subClass CLASS name
          if (subc.getLabel(null) != null) {
            semSeeArrayABC[z][x][0] = subc.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = subc.getLocalName();
          }
          semSeeArrayABC[z][x][1] = subc.getLocalName();
          semSeeArrayABC[z][x][4] = subWt; // Weak/subclass of keyword match
          // FOR IDENT INCL/EXCL/MUST HAVES
          //                      if (boolean exclSUBClass) {
          //                        semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1,2 or 0
          //                      }
          semAnswerTxt3.append("[" + (x + 1) + "] subC: " + semSeeArrayABC[z][x][0] + " wt " +
                               semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
          x++;
        }
      }
      if (doSubCInstanceProc && !superClassSeekingSubCInstances) { //permitSubCInstanceWrapUP) { // removed 11.3.2
        doSubCInstanceProc = false; // RESETS FOR NEXT CLASS/ONTOLOGY MATCH
        // NOW get baseClass INSTANCES [includes all subClasses]
        for (Iterator d = semc.listInstances(); d.hasNext(); ) {
          OntResource semInst = (OntResource) d.next();
          if (semInst.getLabel(null) != null) {
            semSeeArrayABC[z][x][0] = semInst.getLabel(null);
          }
          else {
            semSeeArrayABC[z][x][0] = semInst.getLocalName();
          }
          semSeeArrayABC[z][x][1] = semInst.getLocalName();
          semSeeArrayABC[z][x][4] = insWt; // INSTANCE/Example of keyword match
          // LOOK FOR A USEABLE INSTANCE OR LABEL MATCH to IDENT INCL/EXCL/MUST HAVES
          for (int ti = 0; ti < keywd; ti++) { // FOR EACH ### Rev 7 November 08###
            if (semSeeKeyWord[ti][0].length() > 0 && semSeeKeyWord[ti][0].compareTo("\u22D9") != 0) {
              if ( (semInst.getLocalName().toLowerCase().compareTo(semSeeKeyWord[ti][0].toLowerCase()) == 0) ||
                  (semSeeArrayABC[z][x][0].toLowerCase().compareTo(semSeeKeyWord[ti][0].toLowerCase()) == 0)) {
                semSeeKeyWord[ti][2] = "Y"; // REVISION 13.12.2008
                semSeeArrayABC[z][x][5] = semSeeKeyWord[ti][1]; // value 1, 2 or 0
              }
            }
          }
          semAnswerTxt3.append("[" + (x + 1) + "] subCInst: " + semSeeArrayABC[z][x][0] + " wt " +
                               semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
          x++;
        }
      }
    } // END OF "if (semSeeKeyWord[t].length() > 0)"
  } // END OF KEYWORD ITERATION

  // NOW FLAG FOR RELATION CLASSES SEARCH
  getRelationClasses = true;
}
//////////////////////////////////////////////////////////////////////////////
//////    FOR Ontology BASE CLASS and ALL Ontology CLASSES process ONLY  ///
//////////////////////////////////////////////////////////////////////////////

// For EACH CLASS take the LABEL name or the CLASS name /////////////////
else if (!subCChkBox.isSelected() && !supCChkBox.isSelected()) {
```

Fig. A5. *S+S OQE* Part 5.

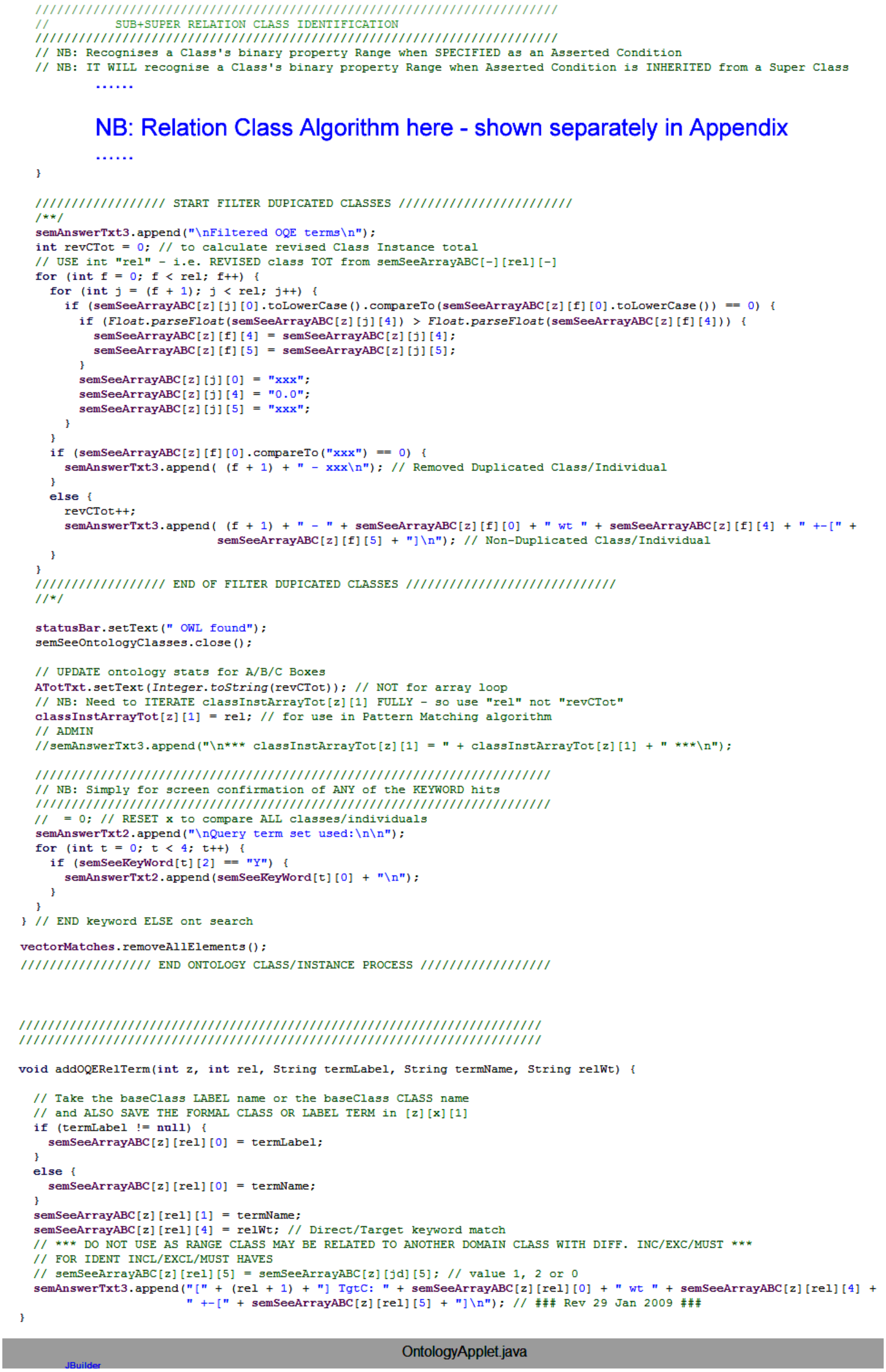
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
    /**/
    // FOR ALL CLASSES Take the Class LABEL name and/or the CLASS name
    // and ALSO SAVE THE FORMAL CLASS OR LABEL TERM in [z][x][1]
    if (semc.getLabel(null) != null) {
      semSeeArrayABC[z][x][0] = semc.getLabel(null);
    }
    else {
      semSeeArrayABC[z][x][0] = semc.getLocalName();
    }
    semSeeArrayABC[z][x][1] = semc.getLocalName();
    semSeeArrayABC[z][x][4] = dirWt; // Direct/Target keyword match
    // LOOP CONTROLS just semSeeArrayABC[z][x][5] Allocation ELSE REPEATED class processing
    for (int t = 0; t < keywd; t++) { // FOR EACH ### Rev 7 November 08 ###
      if (semSeeKeyWord[t][0].length() > 0 && semSeeKeyWord[t][0].compareTo("\u22D9") != 0) {
        //proceedAllClasses = true; // CONTROL FOR MULTI-ONTOLOGIES //REMOVED 10.3.2009***
        // LOOK FOR A USEABLE CLASS OR LABEL MATCH to IDENT INCL/EXCL/MUST HAVES
        if ( (semc.getLocalName().toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0) ||
            (semSeeArrayABC[z][x][0].toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0)) {
          semSeeKeyWord[t][2] = "Y"; // REVISION 13.12.2008
          semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
        }
      }
    }
    semAnswerTxt3.append("[" + (x + 1) + "] OntC: " + semSeeArrayABC[z][x][0] + " wt " + semSeeArrayABC[z][x][4] +
                         " +-[" + semSeeArrayABC[z][x][5] + "]\n");
    x++;
  }
}
//*/


/**/
// ALL ONTOLOGY (and BASE CLASS) INSTANCES
if (!subCChkBox.isSelected() && !supCChkBox.isSelected()) { //&& proceedAllClasses) { //REMOVED 10.3.2009***
  //proceedAllClasses = false; // RESET FOR NEXT ONTOLOGY //REMOVED 10.3.2009***
  //ADMIN Print relations for all ontology
  getRelationClasses = true;
  for (Iterator d = semSeeOntologyClasses.listIndividuals(); d.hasNext(); ) {
    OntResource semInst = (OntResource) d.next();

    if (semInst.getLabel(null) != null) {
      semSeeArrayABC[z][x][0] = semInst.getLabel(null);
    }
    else {
      semSeeArrayABC[z][x][0] = semInst.getLocalName();
    }
    semSeeArrayABC[z][x][1] = semInst.getLocalName();
    semSeeArrayABC[z][x][4] = insWt; // INSTANCE/Example of keyword match
    // LOOP CONTROLS just semSeeArrayABC[z][x][5] Allocation ELSE REPEATED class processing
    for (int t = 0; t < keywd; t++) { // FOR EACH ### Rev 7 November ###
      if (semSeeKeyWord[t][0].length() > 0 && semSeeKeyWord[t][0].compareTo("\u22D9") != 0) {
        // LOOK FOR A USEABLE INSTANCE OR LABEL MATCH to IDENT INCL/EXCL/MUST HAVES
        if ( (semInst.getLocalName().toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0) ||
            (semSeeArrayABC[z][x][0].toLowerCase().compareTo(semSeeKeyWord[t][0].toLowerCase()) == 0)) {
          semSeeKeyWord[t][2] = "Y"; // REVISION 13.12.2008
          semSeeArrayABC[z][x][5] = semSeeKeyWord[t][1]; // value 1, 2 or 0
        }
      }
    }
    semAnswerTxt3.append("[" + (x + 1) + "] OntInst: " + semSeeArrayABC[z][x][0] + " wt " +
                         semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
    x++;
  }
}
//*/

//////////////////////////////////// REV 13.12.2008 ///////////////////////
//                         KEYWORD OQE NON-MATCH MOP-UP                  //
///////////////////////////////////////////////////////////////////////////
boolean missMatch = false;
boolean getheader = true;
for (int tj = 0; tj < keywd; tj++) { // FOR EACH ### Rev 7 November 08###
  if ( (semSeeKeyWord[tj][0].length() > 0 && semSeeKeyWord[tj][0].compareTo("\u22D9") != 0) &&
      (semSeeKeyWord[tj][2].compareTo("N") == 0)) {
    missMatch = true;
    semSeeArrayABC[z][x][0] = semSeeKeyWord[tj][0]; // i.e. same as ONT label
    semSeeArrayABC[z][x][1] = semSeeKeyWord[tj][0]; // i.e. same as ONT class name
    semSeeArrayABC[z][x][4] = dirWt; // Direct/Target keyword match
    // FOR INCL/EXCL/MUST HAVES
    semSeeArrayABC[z][x][5] = semSeeKeyWord[tj][1]; // value 1, 2 or 0
    if (getheader) {
      getheader = false;
      semAnswerTxt2.append("ONT [" + Integer.toString(z + 1) + "] Keyword miss-matches:\n");
    }
    semAnswerTxt2.append("keyword [" + semSeeArrayABC[z][x][0] + "] \u2262 [any " + semContextTxt.getText() +
                         " concept] \u2717\u2717\n *** Keyword added ***\n");
    semSeeKeyWord[tj][2] = "Y";
    semAnswerTxt3.append("\n[" + (x + 1) + "] Keyword non-match: " + semSeeArrayABC[z][x][0] + " wt " +
                         semSeeArrayABC[z][x][4] + " +-[" + semSeeArrayABC[z][x][5] + "]\n");
    x++; // see cell limit ref 16 lines below
  }
}
if (!missMatch) {
  semAnswerTxt2.append("ONT [" + Integer.toString(z + 1) + "] NO keyword miss-matches\n");
}
```

Fig. A6. *S+S OQE* Part 6.

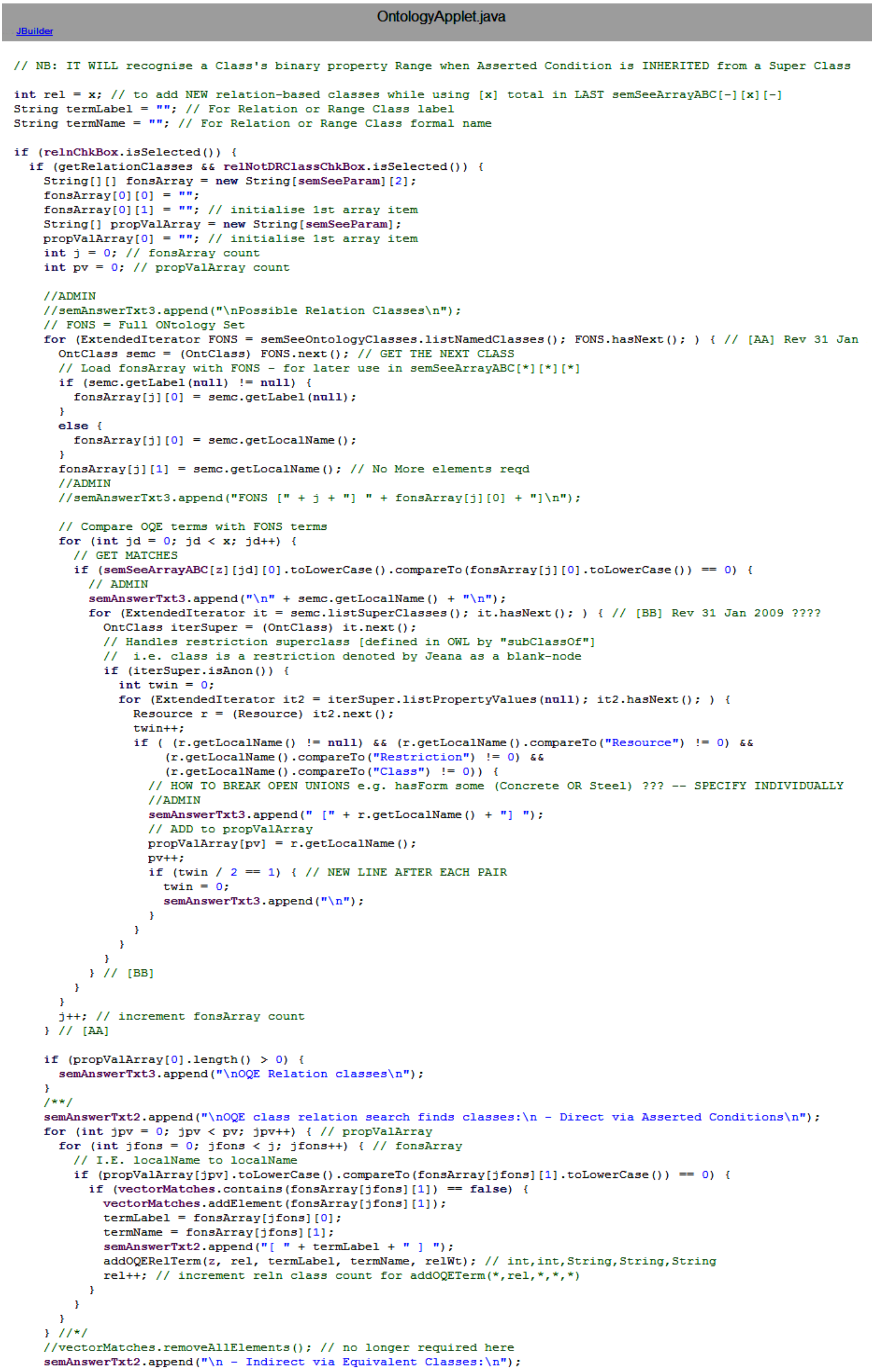
Inheritance Class Hierarchy Algorithm (***S+S OQE*** including Equivalent classes) – contd.



```
  ////////////////////////////////////////////////////////////////////////
  //          SUB+SUPER RELATION CLASS IDENTIFICATION
  ////////////////////////////////////////////////////////////////////////
  // NB: Recognises a Class's binary property Range when SPECIFIED as an Asserted Condition
  // NB: IT WILL recognise a Class's binary property Range when Asserted Condition is INHERITED from a Super Class
          ......

          NB: Relation Class Algorithm here - shown separately in Appendix

          ......
  }

  ////////////////// START FILTER DUPICATED CLASSES ////////////////////////
  /**/
  semAnswerTxt3.append("\nFiltered OQE terms\n");
  int revCTot = 0; // to calculate revised Class Instance total
  // USE int "rel" - i.e. REVISED class TOT from semSeeArrayABC[-][rel][-]
  for (int f = 0; f < rel; f++) {
    for (int j = (f + 1); j < rel; j++) {
      if (semSeeArrayABC[z][j][0].toLowerCase().compareTo(semSeeArrayABC[z][f][0].toLowerCase()) == 0) {
        if (Float.parseFloat(semSeeArrayABC[z][j][4]) > Float.parseFloat(semSeeArrayABC[z][f][4])) {
          semSeeArrayABC[z][f][4] = semSeeArrayABC[z][j][4];
          semSeeArrayABC[z][f][5] = semSeeArrayABC[z][j][5];
        }
        semSeeArrayABC[z][j][0] = "xxx";
        semSeeArrayABC[z][j][4] = "0.0";
        semSeeArrayABC[z][j][5] = "xxx";
      }
    }
    if (semSeeArrayABC[z][f][0].compareTo("xxx") == 0) {
      semAnswerTxt3.append( (f + 1) + " - xxx\n"); // Removed Duplicated Class/Individual
    }
    else {
      revCTot++;
      semAnswerTxt3.append( (f + 1) + " - " + semSeeArrayABC[z][f][0] + " wt " + semSeeArrayABC[z][f][4] + " +-[" +
                           semSeeArrayABC[z][f][5] + "]\n"); // Non-Duplicated Class/Individual
    }
  }
  ////////////////// END OF FILTER DUPICATED CLASSES /////////////////////////////
  //*/

  statusBar.setText(" OWL found");
  semSeeOntologyClasses.close();

  // UPDATE ontology stats for A/B/C Boxes
  ATotTxt.setText(Integer.toString(revCTot)); // NOT for array loop
  // NB: Need to ITERATE classInstArrayTot[z][1] FULLY - so use "rel" not "revCTot"
  classInstArrayTot[z][1] = rel; // for use in Pattern Matching algorithm
  // ADMIN
  //semAnswerTxt3.append("\n*** classInstArrayTot[z][1] = " + classInstArrayTot[z][1] + " ***\n");

  ////////////////////////////////////////////////////////////////////////
  // NB: Simply for screen confirmation of ANY of the KEYWORD hits
  ////////////////////////////////////////////////////////////////////////
  //  = 0; // RESET x to compare ALL classes/individuals
  semAnswerTxt2.append("\nQuery term set used:\n\n");
  for (int t = 0; t < 4; t++) {
    if (semSeeKeyWord[t][2] == "Y") {
      semAnswerTxt2.append(semSeeKeyWord[t][0] + "\n");
    }
  }
} // END keyword ELSE ont search

vectorMatches.removeAllElements();
////////////////// END ONTOLOGY CLASS/INSTANCE PROCESS //////////////////



////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////

void addOQERelTerm(int z, int rel, String termLabel, String termName, String relWt) {

  // Take the baseClass LABEL name or the baseClass CLASS name
  // and ALSO SAVE THE FORMAL CLASS OR LABEL TERM in [z][x][1]
  if (termLabel != null) {
    semSeeArrayABC[z][rel][0] = termLabel;
  }
  else {
    semSeeArrayABC[z][rel][0] = termName;
  }
  semSeeArrayABC[z][rel][1] = termName;
  semSeeArrayABC[z][rel][4] = relWt; // Direct/Target keyword match
  // *** DO NOT USE AS RANGE CLASS MAY BE RELATED TO ANOTHER DOMAIN CLASS WITH DIFF. INC/EXC/MUST ***
  // FOR IDENT INCL/EXCL/MUST HAVES
  // semSeeArrayABC[z][rel][5] = semSeeArrayABC[z][jd][5]; // value 1, 2 or 0
  semAnswerTxt3.append("[" + (rel + 1) + "] TgtC: " + semSeeArrayABC[z][rel][0] + " wt " + semSeeArrayABC[z][rel][4] +
                       " +-[" + semSeeArrayABC[z][rel][5] + "]\n"); // ### Rev 29 Jan 2009 ###
}
```

Fig. A7. *S+S OQE* Part 7.

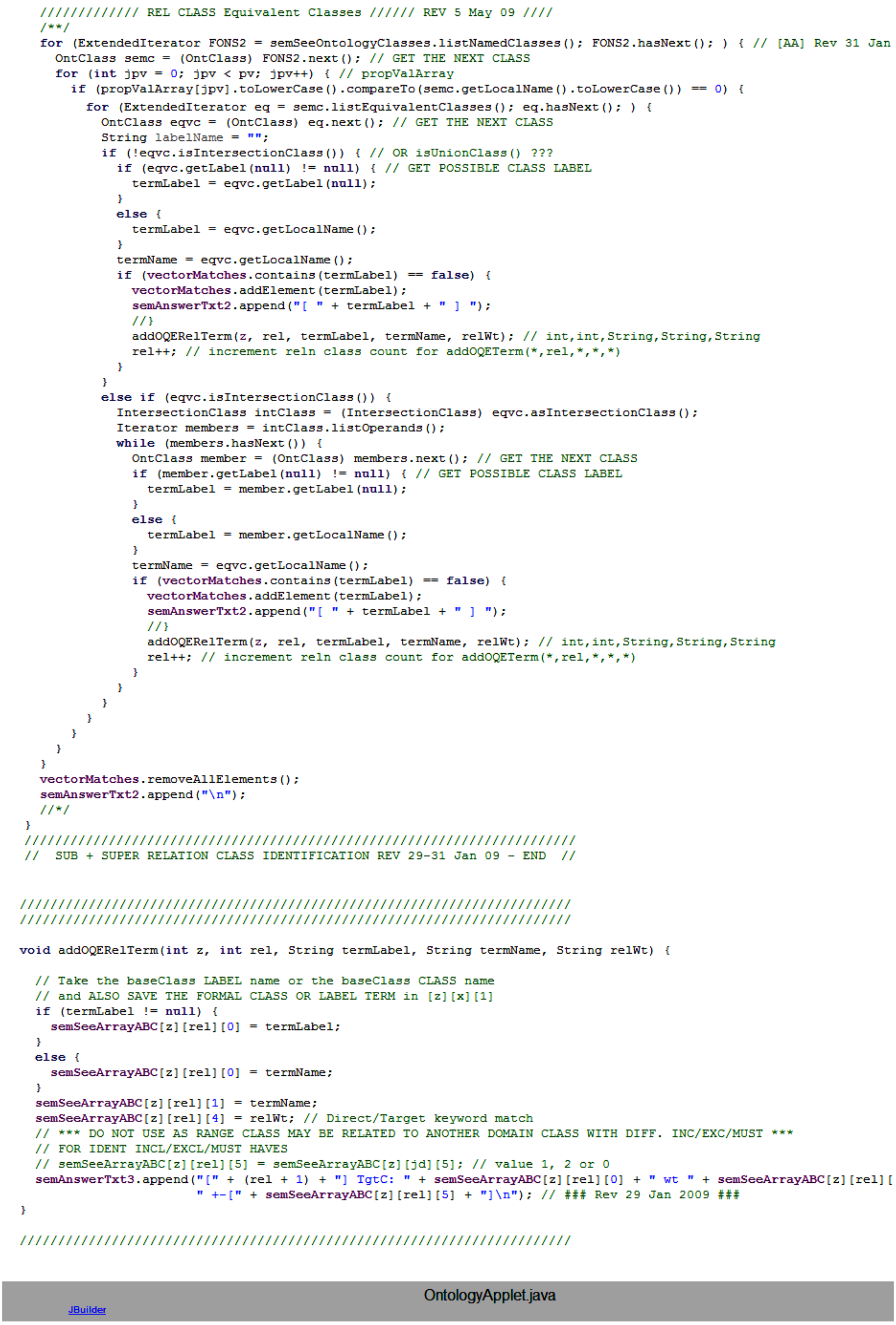
Relation Class Algorithm (*S*+*S*+***R OQE*** stage 1: get **Relation** classes only).



```java
// NB: IT WILL recognise a Class's binary property Range when Asserted Condition is INHERITED from a Super Class

int rel = x; // to add NEW relation-based classes while using [x] total in LAST semSeeArrayABC[-][x][-]
String termLabel = ""; // For Relation or Range Class label
String termName = ""; // For Relation or Range Class formal name

if (relnChkBox.isSelected()) {
  if (getRelationClasses && relNotDRClassChkBox.isSelected()) {
    String[][] fonsArray = new String[semSeeParam][2];
    fonsArray[0][0] = "";
    fonsArray[0][1] = ""; // initialise 1st array item
    String[] propValArray = new String[semSeeParam];
    propValArray[0] = ""; // initialise 1st array item
    int j = 0; // fonsArray count
    int pv = 0; // propValArray count

    //ADMIN
    //semAnswerTxt3.append("\nPossible Relation Classes\n");
    // FONS = Full ONtology Set
    for (ExtendedIterator FONS = semSeeOntologyClasses.listNamedClasses(); FONS.hasNext(); ) { // [AA] Rev 31 Jan
      OntClass semc = (OntClass) FONS.next(); // GET THE NEXT CLASS
      // Load fonsArray with FONS - for later use in semSeeArrayABC[*][*][*]
      if (semc.getLabel(null) != null) {
        fonsArray[j][0] = semc.getLabel(null);
      }
      else {
        fonsArray[j][0] = semc.getLocalName();
      }
      fonsArray[j][1] = semc.getLocalName(); // No More elements reqd
      //ADMIN
      //semAnswerTxt3.append("FONS [" + j + "] " + fonsArray[j][0] + "]\n");

      // Compare OQE terms with FONS terms
      for (int jd = 0; jd < x; jd++) {
        // GET MATCHES
        if (semSeeArrayABC[z][jd][0].toLowerCase().compareTo(fonsArray[j][0].toLowerCase()) == 0) {
          // ADMIN
          semAnswerTxt3.append("\n" + semc.getLocalName() + "\n");
          for (ExtendedIterator it = semc.listSuperClasses(); it.hasNext(); ) { // [BB] Rev 31 Jan 2009 ????
            OntClass iterSuper = (OntClass) it.next();
            // Handles restriction superclass [defined in OWL by "subClassOf"]
            //  i.e. class is a restriction denoted by Jeana as a blank-node
            if (iterSuper.isAnon()) {
              int twin = 0;
              for (ExtendedIterator it2 = iterSuper.listPropertyValues(null); it2.hasNext(); ) {
                Resource r = (Resource) it2.next();
                twin++;
                if ( (r.getLocalName() != null) && (r.getLocalName().compareTo("Resource") != 0) &&
                    (r.getLocalName().compareTo("Restriction") != 0) &&
                    (r.getLocalName().compareTo("Class") != 0)) {
                  // HOW TO BREAK OPEN UNIONS e.g. hasForm some (Concrete OR Steel) ??? -- SPECIFY INDIVIDUALLY
                  //ADMIN
                  semAnswerTxt3.append(" [" + r.getLocalName() + "] ");
                  // ADD to propValArray
                  propValArray[pv] = r.getLocalName();
                  pv++;
                  if (twin / 2 == 1) { // NEW LINE AFTER EACH PAIR
                    twin = 0;
                    semAnswerTxt3.append("\n");
                  }
                }
              }
            }
          } // [BB]
        }
      }
      j++; // increment fonsArray count
    } // [AA]

    if (propValArray[0].length() > 0) {
      semAnswerTxt3.append("\nOQE Relation classes\n");
    }
    /**/
    semAnswerTxt2.append("\nOQE class relation search finds classes:\n - Direct via Asserted Conditions\n");
    for (int jpv = 0; jpv < pv; jpv++) { // propValArray
      for (int jfons = 0; jfons < j; jfons++) { // fonsArray
        // I.E. localName to localName
        if (propValArray[jpv].toLowerCase().compareTo(fonsArray[jfons][1].toLowerCase()) == 0) {
          if (vectorMatches.contains(fonsArray[jfons][1]) == false) {
            vectorMatches.addElement(fonsArray[jfons][1]);
            termLabel = fonsArray[jfons][0];
            termName = fonsArray[jfons][1];
            semAnswerTxt2.append("[ " + termLabel + " ] ");
            addOQERelTerm(z, rel, termLabel, termName, relWt); // int,int,String,String,String
            rel++; // increment reln class count for addOQETerm(*,rel,*,*,*)
          }
        }
      }
    } //*/
    //vectorMatches.removeAllElements(); // no longer required here
    semAnswerTxt2.append("\n - Indirect via Equivalent Classes:\n");
```

Fig. A8. *S*+*S*+***R OQE*** stage 1.

Relation Class Algorithm (*S*+*S*+***R OQE*** stage 2: get ***R*** class **Equivalent** classes).



```
  ///////////// REL CLASS Equivalent Classes ////// REV 5 May 09 ////
  /**/
  for (ExtendedIterator FONS2 = semSeeOntologyClasses.listNamedClasses(); FONS2.hasNext(); ) { // [AA] Rev 31 Jan
    OntClass semc = (OntClass) FONS2.next(); // GET THE NEXT CLASS
    for (int jpv = 0; jpv < pv; jpv++) { // propValArray
      if (propValArray[jpv].toLowerCase().compareTo(semc.getLocalName().toLowerCase()) == 0) {
        for (ExtendedIterator eq = semc.listEquivalentClasses(); eq.hasNext(); ) {
          OntClass eqvc = (OntClass) eq.next(); // GET THE NEXT CLASS
          String labelName = "";
          if (!eqvc.isIntersectionClass()) { // OR isUnionClass() ???
            if (eqvc.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
              termLabel = eqvc.getLabel(null);
            }
            else {
              termLabel = eqvc.getLocalName();
            }
            termName = eqvc.getLocalName();
            if (vectorMatches.contains(termLabel) == false) {
              vectorMatches.addElement(termLabel);
              semAnswerTxt2.append("[ " + termLabel + " ] ");
              //}
              addOQERelTerm(z, rel, termLabel, termName, relWt); // int,int,String,String,String
              rel++; // increment reln class count for addOQETerm(*,rel,*,*,*)
            }
          }
          else if (eqvc.isIntersectionClass()) {
            IntersectionClass intClass = (IntersectionClass) eqvc.asIntersectionClass();
            Iterator members = intClass.listOperands();
            while (members.hasNext()) {
              OntClass member = (OntClass) members.next(); // GET THE NEXT CLASS
              if (member.getLabel(null) != null) { // GET POSSIBLE CLASS LABEL
                termLabel = member.getLabel(null);
              }
              else {
                termLabel = member.getLocalName();
              }
              termName = eqvc.getLocalName();
              if (vectorMatches.contains(termLabel) == false) {
                vectorMatches.addElement(termLabel);
                semAnswerTxt2.append("[ " + termLabel + " ] ");
                //}
                addOQERelTerm(z, rel, termLabel, termName, relWt); // int,int,String,String,String
                rel++; // increment reln class count for addOQETerm(*,rel,*,*,*)
              }
            }
          }
        }
      }
    }
  }
  vectorMatches.removeAllElements();
  semAnswerTxt2.append("\n");
  //*/
}
////////////////////////////////////////////////////////////////////////
//  SUB + SUPER RELATION CLASS IDENTIFICATION REV 29-31 Jan 09 - END  //


////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////

void addOQERelTerm(int z, int rel, String termLabel, String termName, String relWt) {

  // Take the baseClass LABEL name or the baseClass CLASS name
  // and ALSO SAVE THE FORMAL CLASS OR LABEL TERM in [z][x][1]
  if (termLabel != null) {
    semSeeArrayABC[z][rel][0] = termLabel;
  }
  else {
    semSeeArrayABC[z][rel][0] = termName;
  }
  semSeeArrayABC[z][rel][1] = termName;
  semSeeArrayABC[z][rel][4] = relWt; // Direct/Target keyword match
  // *** DO NOT USE AS RANGE CLASS MAY BE RELATED TO ANOTHER DOMAIN CLASS WITH DIFF. INC/EXC/MUST ***
  // FOR IDENT INCL/EXCL/MUST HAVES
  // semSeeArrayABC[z][rel][5] = semSeeArrayABC[z][jd][5]; // value 1, 2 or 0
  semAnswerTxt3.append("[" + (rel + 1) + "] TgtC: " + semSeeArrayABC[z][rel][0] + " wt " + semSeeArrayABC[z][rel][
                      " +-[" + semSeeArrayABC[z][rel][5] + "]\n"); // ### Rev 29 Jan 2009 ###
}

////////////////////////////////////////////////////////////////////////
```

OntologyApplet.java

JBuilder

Fig. A9. *S*+*S*+***R OQE*** stage 2.

APPENDIX C: ONTOLOGY CONTEXTS USED IN EXPERIMENTS

This section contains graphical representations of the following ontologies:

1. T401 Immigration ontology context – page XXI.

   Fig. A10. T401 Immigration context.

2. T416 Hydro-electric ontology context – page XXII.

   Fig. A11. T416 Hydro-electric context.

3. T438 Tourism Multi-context ontology – pages XXIII and XXIV.

   Fig. A12. T438 Tourism multi-context ontology – part 1.

   Fig. A13. T438 Tourism multi-context ontology – part 2.

4. T401 Immigration with SUMO (super classes) ontology – pages XXV and XXVI.

   Fig. A14. T401 Immigration with SUMO (super classes) ontology – part 1.

   Fig. A15. T401 Immigration with SUMO (super classes) ontology – part 2.

T401 Immigration ontology context

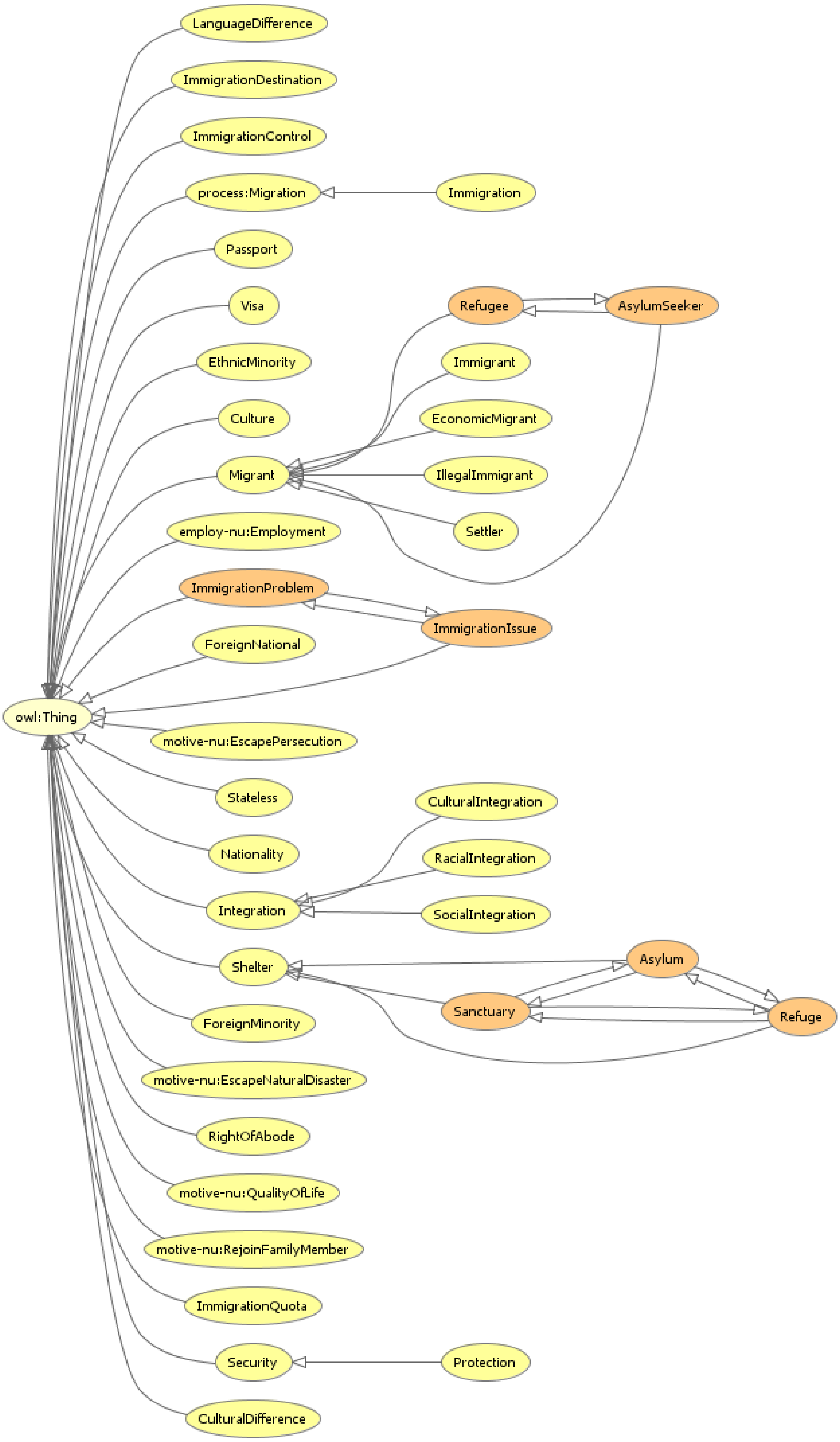


Fig. A10. T401 Immigration context.

T416 Hydro-electric ontology context

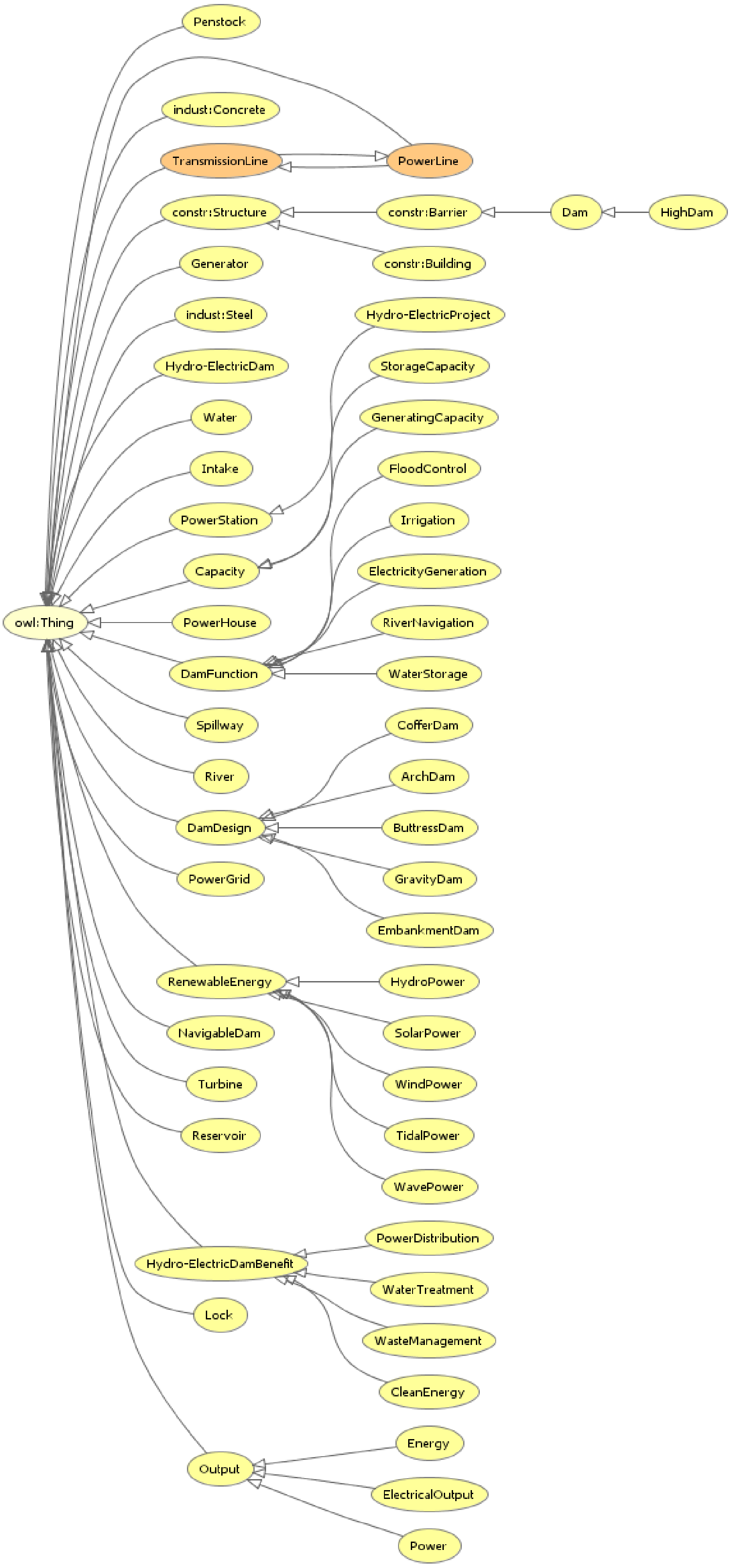


Fig. A11. T416 Hydro-electric context.

T438 Tourism Multi-context ontology (part 1)

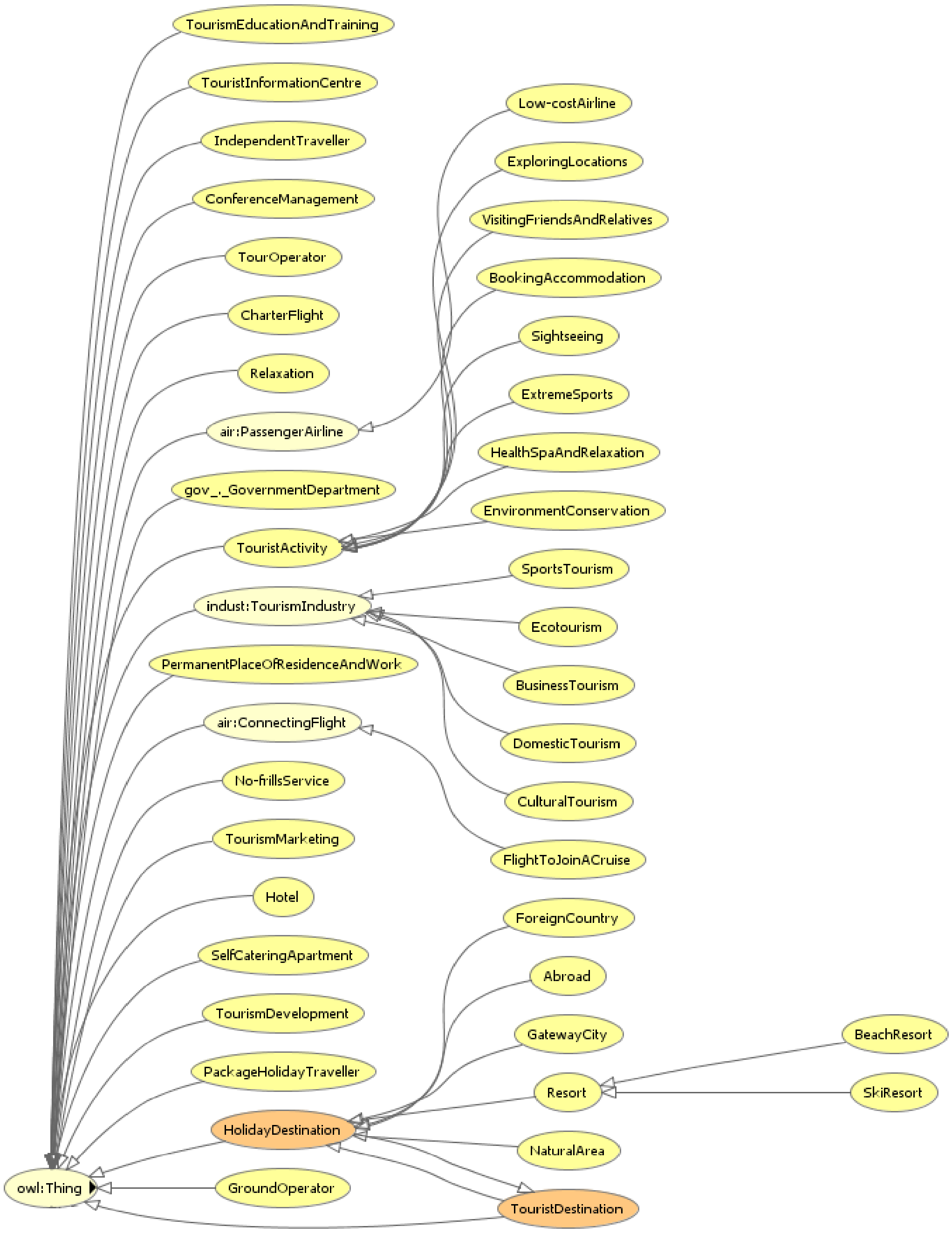


Fig. A12. T438 Tourism multi-context ontology – part 1.

T438 Tourism Multi-context ontology (part 2)

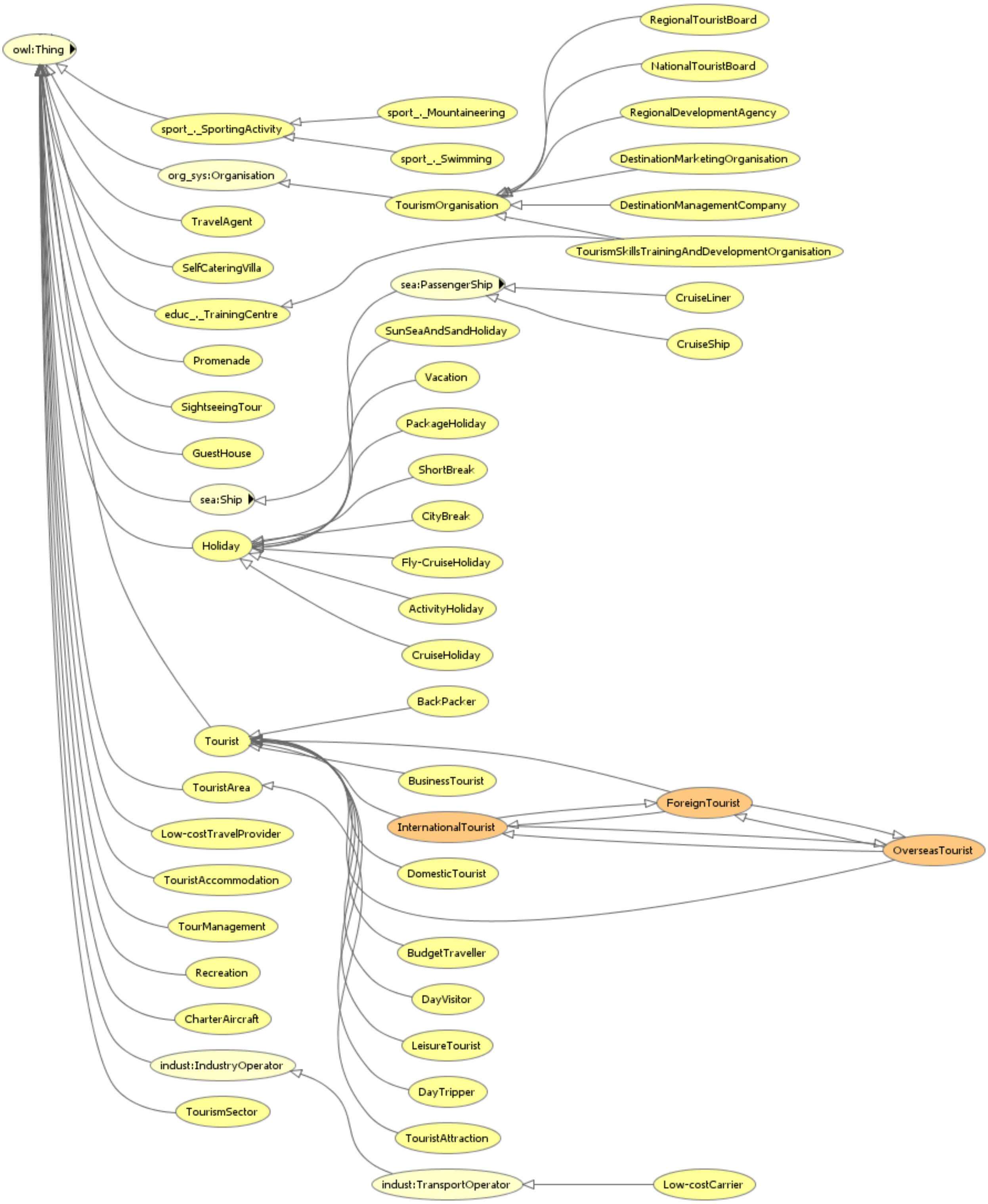


Fig. A13. T438 Tourism multi-context ontology – part 2.

T401 Immigration with SUMO (super classes) ontology (part 1)

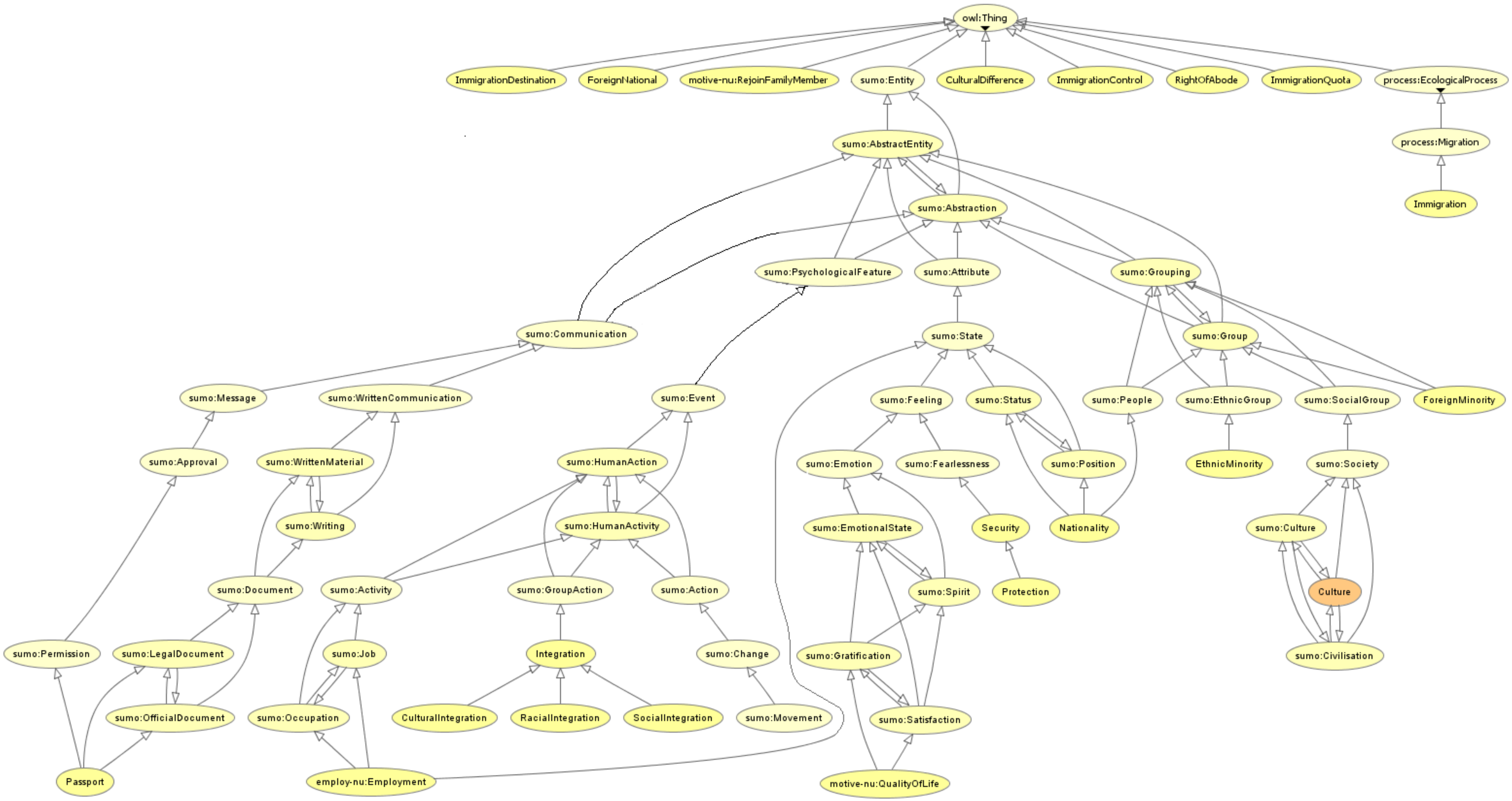


Fig. A14. T401 Immigration with SUMO (super classes) ontology – part 1.

T401 Immigration with SUMO (super classes) ontology (part 2)

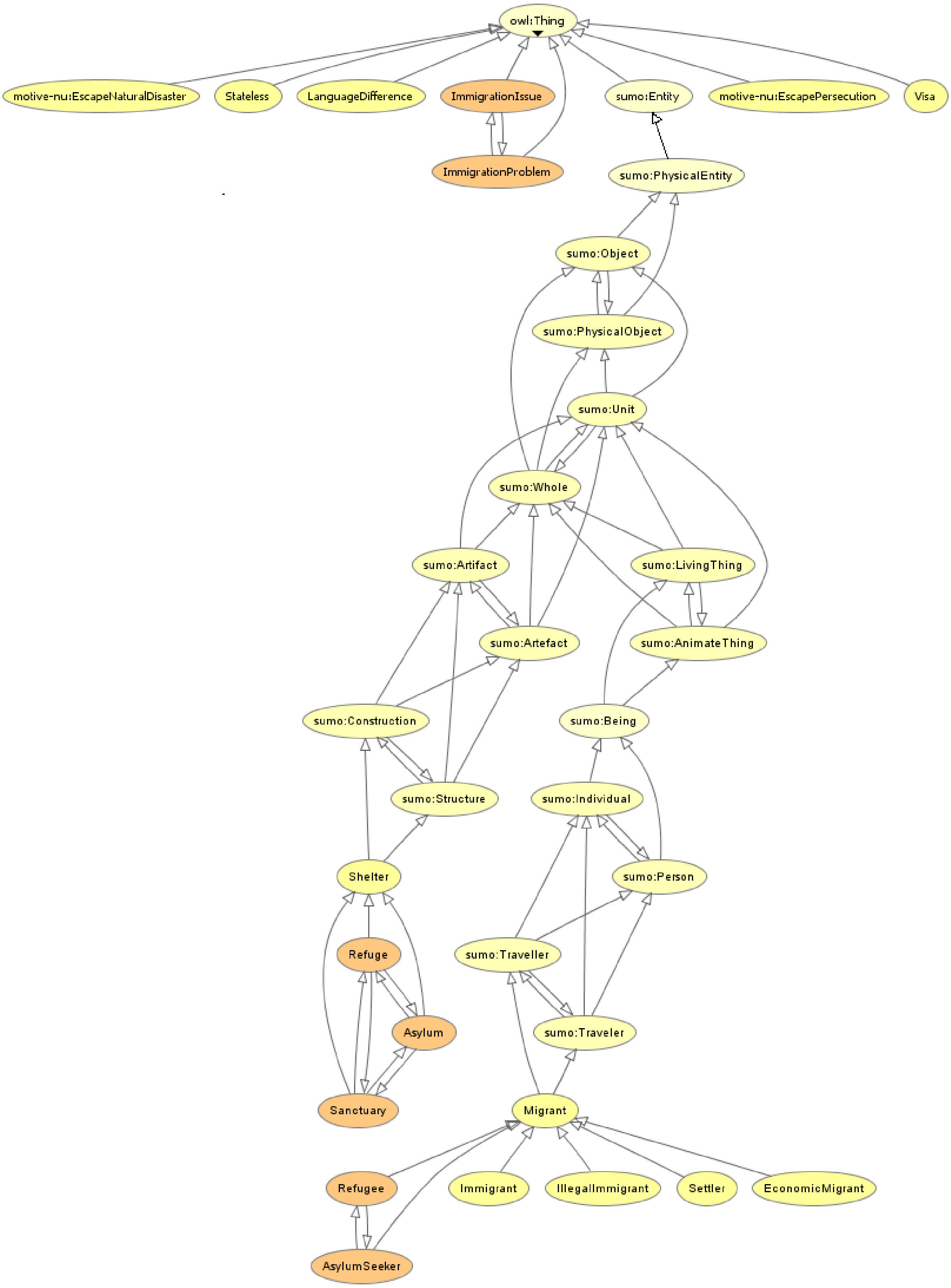


Fig. A15. T401 Immigration with SUMO (super classes) ontology – part 2.

APPENDIX D: VECTOR SPACE MODEL *TF-IDF* JAVA CODE

This section provides a *tf-idf* Java code extract and an explanation of the main Java code variables that correlate to the term weight vector W_{td} in the modified VSM *tf-idf* algorithm referred to in sections 1.6.2 and 3.2.5, where the term weight vector W_{td} was represented as:

$$W_{td} = \sum_{t_i \in d,\, d \in D} \left(\frac{F_{t_i d} * O_w}{\max F_{td}} \right) * \ln \left(\frac{D}{n_{t_i}} \right)$$

The term weight vector is the product of *normalised term frequency* and *inverse document frequency*, i.e. $\left(\frac{F_{t_i d} * O_w}{\max F_{td}} \right)$ and $\ln \left(\frac{D}{n_{t_i}} \right)$ respectively.

The above components are represented by the following variables, which can be found in the *tf-idf* Java code extract in Fig. A16 - shown overleaf.

Normalised term frequency - Java code variable: **normalisedFreq_td**

normalisedFreq_td = ((Freq_td * classValue) / maxFreq)

where Freq_td = frequency of term in document, classValue = ontology class relevance weight and maxFreq = maximum term frequency in document.

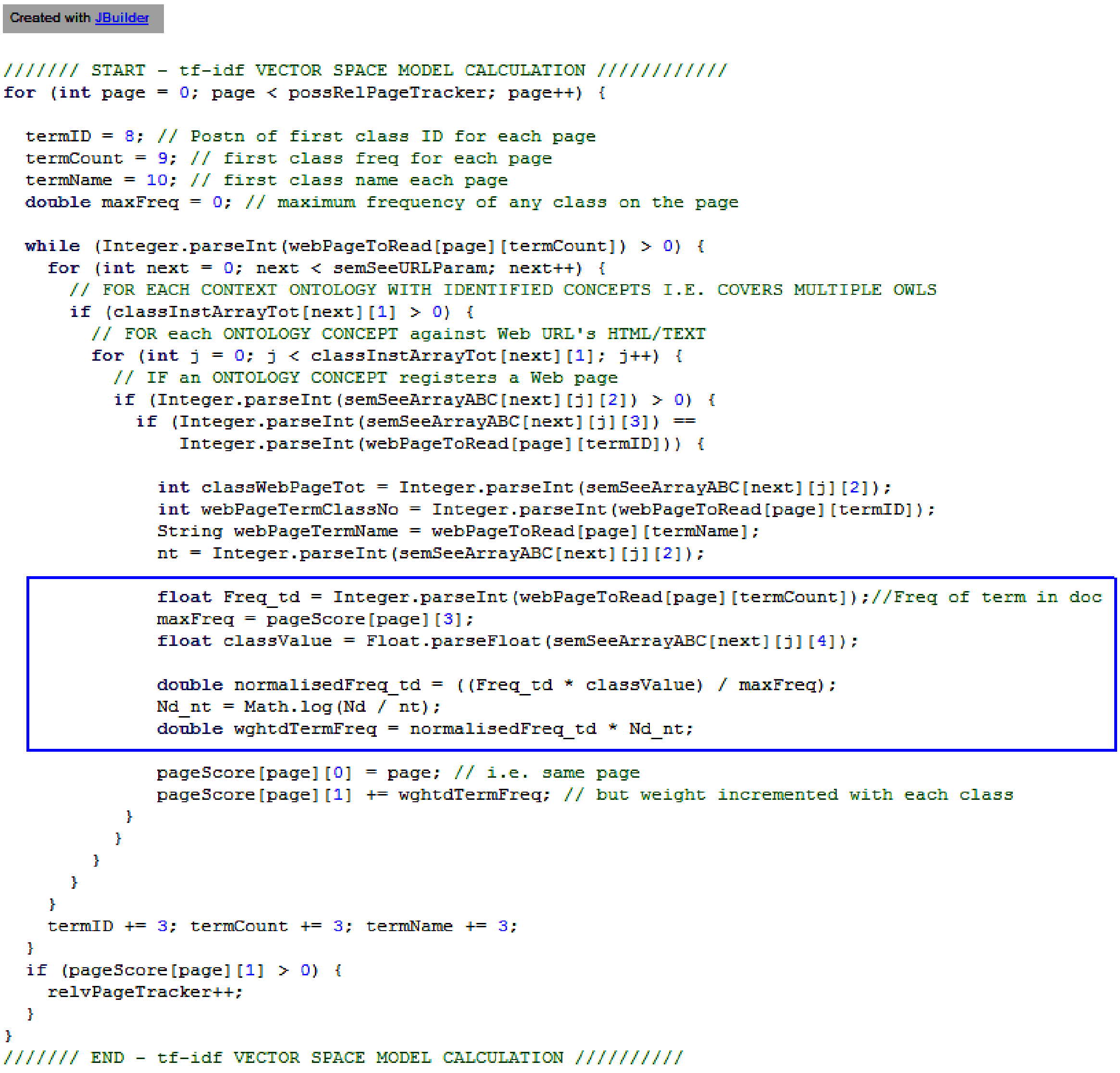
Inverse document frequency - Java code variable: **Nd_nt**

Nd_nt = Math.log (Nd / nt)

where Nd = number of documents in corpus and nt = number of documents containing term.

Term weight vector W_{td} - Java code variable: **wghtdTermFreq**

wghtdTermFreq = normalisedFreq_td * Nd_nt



Created with JBuilder

```
/////// START - tf-idf VECTOR SPACE MODEL CALCULATION ////////////
for (int page = 0; page < possRelPageTracker; page++) {

  termID = 8; // Postn of first class ID for each page
  termCount = 9; // first class freq for each page
  termName = 10; // first class name each page
  double maxFreq = 0; // maximum frequency of any class on the page

  while (Integer.parseInt(webPageToRead[page][termCount]) > 0) {
    for (int next = 0; next < semSeeURLParam; next++) {
      // FOR EACH CONTEXT ONTOLOGY WITH IDENTIFIED CONCEPTS I.E. COVERS MULTIPLE OWLS
      if (classInstArrayTot[next][1] > 0) {
        // FOR each ONTOLOGY CONCEPT against Web URL's HTML/TEXT
        for (int j = 0; j < classInstArrayTot[next][1]; j++) {
          // IF an ONTOLOGY CONCEPT registers a Web page
          if (Integer.parseInt(semSeeArrayABC[next][j][2]) > 0) {
            if (Integer.parseInt(semSeeArrayABC[next][j][3]) ==
                Integer.parseInt(webPageToRead[page][termID])) {

              int classWebPageTot = Integer.parseInt(semSeeArrayABC[next][j][2]);
              int webPageTermClassNo = Integer.parseInt(webPageToRead[page][termID]);
              String webPageTermName = webPageToRead[page][termName];
              nt = Integer.parseInt(semSeeArrayABC[next][j][2]);

              float Freq_td = Integer.parseInt(webPageToRead[page][termCount]);//Freq of term in doc
              maxFreq = pageScore[page][3];
              float classValue = Float.parseFloat(semSeeArrayABC[next][j][4]);

              double normalisedFreq_td = ((Freq_td * classValue) / maxFreq);
              Nd_nt = Math.log(Nd / nt);
              double wghtdTermFreq = normalisedFreq_td * Nd_nt;

              pageScore[page][0] = page; // i.e. same page
              pageScore[page][1] += wghtdTermFreq; // but weight incremented with each class
            }
          }
        }
      }
    }
    termID += 3; termCount += 3; termName += 3;
  }
  if (pageScore[page][1] > 0) {
    relvPageTracker++;
  }
}
/////// END - tf-idf VECTOR SPACE MODEL CALCULATION //////////
```

Fig. A16. *Tf-idf* Java code extract.

APPENDIX E: *OQE* TERM MATCHES FOR MAIN EXPERIMENTS

T401 Experiment (section 4.1)

Query	OQE Mode	Terms	Matches	% Match
Q1	*All*	41	37	90.2
Q2	*All*	41	37	90.2
Q3	*All*	41	37	90.2
Q4	*All*	41	37	90.2
Q5	*All*	41	37	90.2
Q6	*All*	41	37	90.2
Q7	*S+S*	5	5	100.0
Q8	*S+S*	9	9	100.0
Q9	*S+S*	5	5	100.0
Q10	*S+S*	13	13	100.0
	Total	278	254	
T401	Round Ave	28	25	**89%**

T416 Experiment (section 4.2)

Query	OQE Mode	Terms	Matches	% Match		Query	OQE Mode	Terms	Matches	% Match
Q1	*S+S*	9	9	100.0		Q1	*S+S+R*	28	26	92.9
Q2	*S+S*	7	7	100.0		Q2	*S+S+R*	19	19	100.0
Q3	*S+S*	10	10	100.0		Q3	*S+S+R*	29	27	93.1
Q4	*S+S*	6	6	100.0		Q4	*S+S+R*	18	18	100.0
Q5	*S+S*	7	7	100.0		Q5	*S+S+R*	19	19	100.0
Q6	*S+S*	7	6	85.7		Q6	*S+S+R*	21	20	95.2
Q7	*S+S*	10	10	100.0		Q7	*S+S+R*	29	27	93.1
Q8	*S+S*	7	7	100.0		Q8	*S+S+R*	19	19	100.0
Q9	*S+S*	8	6	75.0		Q9	*S+S+R*	19	17	89.5
Q10	*S+S*	6	4	66.7		Q10	*S+S+R*	7	5	71.4
	Total	77	72				Total	208	197	
T416	Round Ave	8	7	**88%**		T416	Round Ave	21	20	**95%**

T438 Experiment 1 (section 4.3)

Query	OQE Mode	Terms	Matches	% Match
Q1	*S+S*	14	12	85.7
Q2	*S+S*	29	21	72.4
Q3	*S+S*	21	15	71.4
Q4	*S+S*	34	24	70.6
Q5	*S+S*	20	15	75.0
Q6	*S+S*	13	13	100.0
Q7	*S+S*	22	13	59.1
Q8	*S+S*	17	15	88.2
Q9	*S+S*	12	10	83.3
Q10	*S+S*	8	8	100.0
Q11	*S+S*	16	13	81.3
Q12	*S+S*	17	11	64.7
Q13	*S+S*	18	15	83.3
Q14	*S+S*	5	5	100.0
Q15	*S+S*	17	14	82.4
Q16	*S+S*	5	5	100.0
Q17	*S+S*	15	13	86.7
Q18	*S+S*	19	15	78.9
Q19	*S+S*	21	18	85.7
Q20	*S+S*	6	6	100.0
	Total	329	261	
T438 (1)	Round Ave	16	13	**81%**

T438 Experiment 2 (section 4.4)

Query	OQE Mode	Terms	Matches	% Match
Q1	*S+S+R*	39	31	79.5
Q2	*S+S+R*	56	41	73.2
Q3	*S+S+R*	31	20	64.5
Q4	*S+S+R*	64	47	73.4
Q5	*S+S+R*	42	34	81.0
Q7	*S+S+R*	36	21	58.3
Q8	*S+S+R*	45	39	86.7
Q11	*S+S+R*	27	20	74.1
Q12	*S+S+R*	28	20	71.4
Q15	*S+S+R*	29	22	75.9
Q16	*S+S+R*	9	9	100.0
Q17	*S+S+R*	39	35	89.7
Q19	*S+S+R*	34	27	79.4
	Total	479	366	
(13 queries)				
T438 (2)	Round Ave	37	28	**76%**

APPENDIX F: PRECISION & RECALL DATA (T401, T416, T438)

Precision and recall data for section 4.1, 4.2, 4.3 and 4.4 experiments is organised as follows.

1. T401 *optional* and *must-have* P&R data by query - pages XXXI to XXXV.
2. T416 *optional* P&R data by query - pages XXXVI to XXXIX.
3. T416 *must-have* P&R data by query - pages XXXIX to XLII.
4. T438 *optional* and *must-have* P&R data by query - pages XLIII to LII.
5. T401 versus T401+SUMO P&R data – pages LIII to LVI.

T401 *Optional* and *Must-have* Queries

T401 Query 1															
X OR Y				X OR Y…n (41)				Must "Germany"				Must "Germany"			
<keyword only> "foreign minority", Germany, culture, integration				<immigration.owl - ALL 41 classes>				<keyword only> "foreign minority", Germany, culture, integration				<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	1448			Retd	3937			Retd	533 (1448)			Retd	533 (3937)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	140	2.9%	11%	4	7	57.1%	11%	4	104	3.8%	11%	4	5	80.0%	11%
7	151	4.6%	19%	7	10	70.0%	19%	7	113	6.2%	19%	7	8	87.5%	19%
11	165	6.7%	30%	11	15	73.3%	30%	11	123	8.9%	30%	11	12	91.7%	30%
15	177	8.5%	41%	15	20	75.0%	41%	15	134	11.2%	41%	15	17	88.2%	41%
19	198	9.6%	51%	19	31	61.3%	51%	19	151	12.6%	51%	19	23	82.6%	51%
22	203	10.8%	59%	22	36	61.1%	59%	22	156	14.1%	59%	22	27	81.5%	59%
26	210	12.4%	70%	26	59	44.1%	70%	26	163	16.0%	70%	26	37	70.3%	70%
30	525	5.7%	81%	30	104	28.8%	81%	30	169	17.8%	81%	30	55	54.5%	81%
33	529	6.2%	89%	33	144	22.9%	89%	33	173	19.1%	89%	33	68	48.5%	89%
37	799	4.6%	100%	37	1415	2.6%	100%	37	443	8.4%	100%	37	179	20.7%	100%

T401 Query 2															
X OR Y				X OR Y…n (41)				Must "Germany"				Must "Germany"			
<keyword only> "ethnic minority", "cultural difference", "immigration issue", Germany				<immigration.owl - ALL 41 classes>				<keyword only>"ethnic minority", "cultural difference", "immigration issue", Germany				<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	568			Retd	3937			Retd	533 (568)			Retd	533 (3937)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	49	8.2%	11%	4	7	57.1%	11%	4	14	28.6%	11%	4	5	80.0%	11%
7	52	13.5%	19%	7	10	70.0%	19%	7	17	41.2%	19%	7	8	87.5%	19%
11	57	19.3%	30%	11	15	73.3%	30%	11	22	50.0%	30%	11	12	91.7%	30%
15	61	24.6%	41%	15	20	75.0%	41%	15	26	57.7%	41%	15	17	88.2%	41%
19	370	5.1%	51%	19	31	61.3%	51%	19	335	5.7%	51%	19	23	82.6%	51%
22	385	5.7%	59%	22	36	61.1%	59%	22	350	6.3%	59%	22	27	81.5%	59%
26	389	6.7%	70%	26	59	44.1%	70%	26	354	7.3%	70%	26	37	70.3%	70%
30	415	7.2%	81%	30	104	28.8%	81%	30	380	7.9%	81%	30	55	54.5%	81%
33	419	7.9%	89%	33	144	22.9%	89%	33	384	8.6%	89%	33	68	48.5%	89%
37	425	8.7%	100%	37	1415	2.6%	100%	37	390	9.5%	100%	37	179	20.7%	100%

T401 Query 3

X OR Y			
<keyword only> migrant, "cultural integration", protection, Germany			
Relv Doc	37	Tot Doc	13065
Retd	1688		
10% Rels	Rtd	P	R
4	20	20.0%	11%
7	23	30.4%	19%
11	35	31.4%	30%
15	39	38.5%	41%
19	51	37.3%	51%
22	87	25.3%	59%
26	94	27.7%	70%
30	103	29.1%	81%
33	167	19.8%	89%
37	482	7.7%	100%

X OR Y…n (41)			
<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065
Retd	3937		
10% Rels	Rtd	P	R
4	7	57.1%	11%
7	10	70.0%	19%
11	15	73.3%	30%
15	20	75.0%	41%
19	31	61.3%	51%
22	36	61.1%	59%
26	59	44.1%	70%
30	104	28.8%	81%
33	144	22.9%	89%
37	1415	2.6%	100%

Must "Germany" X OR Y			
<keyword only> migrant, "cultural integration", protection, Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (1688)		
10% Rels	Rtd	P	R
4	17	23.5%	11%
7	20	35.0%	19%
11	24	45.8%	30%
15	28	53.6%	41%
19	38	50.0%	51%
22	73	30.1%	59%
26	80	32.5%	70%
30	89	33.7%	81%
33	106	31.1%	89%
37	418	8.9%	100%

Must "Germany"			
<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065
Retd	533 (3937)		
10% Rels	Rtd	P	R
4	5	80.0%	11%
7	8	87.5%	19%
11	12	91.7%	30%
15	17	88.2%	41%
19	23	82.6%	51%
22	27	81.5%	59%
26	37	70.3%	70%
30	55	54.5%	81%
33	68	48.5%	89%
37	179	20.7%	100%

T401 Query 4

X OR Y			
<keyword only> "foreign minority", immigration, refugee, Germany			
Relv Doc	37	Tot Doc	13065
Retd	708		
10% Rels	Rtd	P	R
4	7	57.1%	11%
7	22	31.8%	19%
11	26	42.3%	30%
15	32	46.9%	41%
19	39	48.7%	51%
22	45	48.9%	59%
26	55	47.3%	70%
30	64	46.9%	81%
33	73	45.2%	89%
37	136	27.2%	100%

X OR Y…n (41)			
<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065
Retd	3937		
10% Rels	Rtd	P	R
4	7	57.1%	11%
7	10	70.0%	19%
11	15	73.3%	30%
15	20	75.0%	41%
19	31	61.3%	51%
22	36	61.1%	59%
26	59	44.1%	70%
30	104	28.8%	81%
33	144	22.9%	89%
37	1415	2.6%	100%

Must "Germany" X OR Y			
<keyword only> "foreign minority", immigration, refugee, Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (708)		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	21	33.3%	19%
11	25	44.0%	30%
15	30	50.0%	41%
19	37	51.4%	51%
22	42	52.4%	59%
26	51	51.0%	70%
30	58	51.7%	81%
33	66	50.0%	89%
37	96	38.5%	100%

Must "Germany"			
<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065
Retd	533 (3937)		
10% Rels	Rtd	P	R
4	5	80.0%	11%
7	8	87.5%	19%
11	12	91.7%	30%
15	17	88.2%	41%
19	23	82.6%	51%
22	27	81.5%	59%
26	37	70.3%	70%
30	55	54.5%	81%
33	68	48.5%	89%
37	179	20.7%	100%

T401 Query 5															
X OR Y				X OR Y…n (41)				Must "Germany" X OR Y				Must "Germany"			
<keyword only> "asylum seeker", employment, "foreign national", Germany				<immigration.owl - ALL 41 classes>				<keyword only> "asylum seeker", employment, "foreign national", Germany				<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	1160			Retd	3937			Retd	533 (1160)			Retd	533 (3937)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	18	22.2%	11%	4	7	57.1%	11%	4	14	28.6%	11%	4	5	80.0%	11%
7	22	31.8%	19%	7	10	70.0%	19%	7	18	38.9%	19%	7	8	87.5%	19%
11	59	18.6%	30%	11	15	73.3%	30%	11	43	25.6%	30%	11	12	91.7%	30%
15	63	23.8%	41%	15	20	75.0%	41%	15	47	31.9%	41%	15	17	88.2%	41%
19	73	26.0%	51%	19	31	61.3%	51%	19	53	35.8%	51%	19	23	82.6%	51%
22	76	28.9%	59%	22	36	61.1%	59%	22	56	39.3%	59%	22	27	81.5%	59%
26	83	31.3%	70%	26	59	44.1%	70%	26	63	41.3%	70%	26	37	70.3%	70%
30	105	28.6%	81%	30	104	28.8%	81%	30	85	35.3%	81%	30	55	54.5%	81%
33	119	27.7%	89%	33	144	22.9%	89%	33	99	33.3%	89%	33	68	48.5%	89%
37	410	9.0%	100%	37	1415	2.6%	100%	37	390	9.5%	100%	37	179	20.7%	100%

T401 Query 6															
X OR Y				X OR Y…n (41)				Must "Germany" X OR Y				Must "Germany"			
<keyword only> migration, "immigration issue", culture, Germany				<immigration.owl - ALL 41 classes>				<keyword only> migration, "immigration issue", culture, Germany				<immigration.owl - ALL 41 classes>			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	1350			Retd	3937			Retd	533 (1350)			Retd	533 (3937)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	6	66.7%	11%	4	7	57.1%	11%	4	6	66.7%	11%	4	5	80.0%	11%
7	12	58.3%	19%	7	10	70.0%	19%	7	12	58.3%	19%	7	8	87.5%	19%
11	16	68.8%	30%	11	15	73.3%	30%	11	16	68.8%	30%	11	12	91.7%	30%
15	24	62.5%	41%	15	20	75.0%	41%	15	20	75.0%	41%	15	17	88.2%	41%
19	59	32.2%	51%	19	31	61.3%	51%	19	54	35.2%	51%	19	23	82.6%	51%
22	64	34.4%	59%	22	36	61.1%	59%	22	59	37.3%	59%	22	27	81.5%	59%
26	83	31.3%	70%	26	59	44.1%	70%	26	77	33.8%	70%	26	37	70.3%	70%
30	101	29.7%	81%	30	104	28.8%	81%	30	94	31.9%	81%	30	55	54.5%	81%
33	120	27.5%	89%	33	144	22.9%	89%	33	111	29.7%	89%	33	68	48.5%	89%
37	395	9.4%	100%	37	1415	2.6%	100%	37	171	21.6%	100%	37	179	20.7%	100%

T401 Query 7

X OR Y			
<keyword only> asylum, immigrant, "quality of life", Germany			
Relv Doc	37	Tot Doc	13065
Retd	826		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	11	63.6%	19%
11	40	27.5%	30%
15	50	30.0%	41%
19	57	33.3%	51%
22	62	35.5%	59%
26	72	36.1%	70%
30	95	31.6%	81%
33	109	30.3%	89%
37	403	9.2%	100%

X OR Y…n (5)			
<immigration.owl 5 S+S classes> from: asylum, immigrant, "quality of life", Germany			
Relv Doc	37	Tot Doc	13065
Retd	851		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	11	63.6%	19%
11	21	52.4%	30%
15	33	45.5%	41%
19	53	35.8%	51%
22	63	34.9%	59%
26	69	37.7%	70%
30	91	33.0%	81%
33	126	26.2%	89%
37	409	9.0%	100%

Must "Germany" X OR Y			
<keyword only> asylum, immigrant, "quality of life", Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (826)		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	9	77.8%	19%
11	36	30.6%	30%
15	46	32.6%	41%
19	52	36.5%	51%
22	57	38.6%	59%
26	66	39.4%	70%
30	83	36.1%	81%
33	93	35.5%	89%
37	110	33.6%	100%

Must "Germany" X OR Y…n (5)			
<immigration.owl 5 S+S classes> from: asylum, immigrant, "quality of life", Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (851)		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	10	70.0%	19%
11	17	64.7%	30%
15	27	55.6%	41%
19	46	41.3%	51%
22	53	41.5%	59%
26	58	44.8%	70%
30	73	41.1%	81%
33	94	35.1%	89%
37	116	31.9%	100%

T401 Query 8

X OR Y			
<keyword only> "asylum seeker", security, shelter, Germany			
Relv Doc	37	Tot Doc	13065
Retd	1844		
10% Rels	Rtd	P	R
4	18	22.2%	11%
7	25	28.0%	19%
11	60	18.3%	30%
15	64	23.4%	41%
19	141	13.5%	51%
22	153	14.4%	59%
26	171	15.2%	70%
30	187	16.0%	81%
33	211	15.6%	89%
37	480	7.7%	100%

X OR Y…n (9)			
<immigration.owl 9 S+S classes> from: "asylum seeker", security, shelter, Germany			
Relv Doc	37	Tot Doc	13065
Retd	2645		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	12	58.3%	19%
11	17	64.7%	30%
15	22	68.2%	41%
19	27	70.4%	51%
22	37	59.5%	59%
26	50	52.0%	70%
30	68	44.1%	81%
33	99	33.3%	89%
37	430	8.6%	100%

Must "Germany"			
<keyword only> "asylum seeker", security, shelter, Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (1844)		
10% Rels	Rtd	P	R
4	13	30.8%	11%
7	18	38.9%	19%
11	48	22.9%	30%
15	52	28.8%	41%
19	62	30.6%	51%
22	70	31.4%	59%
26	88	29.5%	70%
30	103	29.1%	81%
33	126	26.2%	89%
37	391	9.5%	100%

Must "Germany" X OR Y…n (9)			
<immigration.owl 9 S+S classes> from: "asylum seeker", security, shelter, Germany			
Relv Doc	37	Tot Doc	13065
Retd	533 (2645)		
10% Rels	Rtd	P	R
4	6	66.7%	11%
7	12	58.3%	19%
11	17	64.7%	30%
15	22	68.2%	41%
19	27	70.4%	51%
22	36	61.1%	59%
26	47	55.3%	70%
30	53	56.6%	81%
33	68	48.5%	89%
37	190	19.5%	100%

T401 Query 9															
X OR Y				X OR Y…n (5)				Must "Germany"				Must "Germany" X OR Y…n (5)			
<keyword only> "economic migrant", "illegal immigrant", "immigration control", Germany				<immigration.owl 5 S+S classes> from: "economic migrant", "illegal immigrant", "immigration control", Germany				<keyword only> "economic migrant", "illegal immigrant", "immigration control", Germany				<immigration.owl 5 S+S classes> from: "economic migrant", "illegal immigrant", "immigration control", Germany			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	559			Retd	612			Retd	533 (559)			Retd	533 (612)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	14	28.6%	11%	4	7	57.1%	11%	4	12	33.3%	11%	4	6	66.7%	11%
7	20	35.0%	19%	7	12	58.3%	19%	7	17	41.2%	19%	7	10	70.0%	19%
11	29	37.9%	30%	11	18	61.1%	30%	11	25	44.0%	30%	11	15	73.3%	30%
15	33	45.5%	41%	15	32	46.9%	41%	15	29	51.7%	41%	15	25	60.0%	41%
19	42	45.2%	51%	19	38	50.0%	51%	19	36	52.8%	51%	19	31	61.3%	51%
22	46	47.8%	59%	22	42	52.4%	59%	22	40	55.0%	59%	22	35	62.9%	59%
26	71	36.6%	70%	26	61	42.6%	70%	26	45	57.8%	70%	26	50	52.0%	70%
30	76	39.5%	81%	30	69	43.5%	81%	30	50	60.0%	81%	30	56	53.6%	81%
33	79	41.8%	89%	33	86	38.4%	89%	33	53	62.3%	89%	33	60	55.0%	89%
37	401	9.2%	100%	37	454	8.1%	100%	37	375	9.9%	100%	37	375	9.9%	100%

T401 Query 10															
X OR Y				X OR Y…n (13)				Must "Germany"				Must "Germany" X OR Y…n (13)			
<keyword only> "cultural difference", integration, migrant, Germany				<immigration.owl 13 S+S classes> from: "cultural difference", integration, migrant, Germany				<keyword only> "cultural difference", integration, migrant, Germany				<immigration.owl 13 S+S classes> from: "cultural difference", integration, migrant, Germany			
Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065	Relv Doc	37	Tot Doc	13065
Retd	952			Retd	1092			Retd	533 (952)			Retd	533 (1092)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	11	36.4%	11%	4	34	11.8%	11%	4	8	50.0%	11%	4	24	16.7%	11%
7	25	28.0%	19%	7	53	13.2%	19%	7	21	33.3%	19%	7	43	16.3%	19%
11	32	34.4%	30%	11	80	13.8%	30%	11	27	40.7%	30%	11	50	22.0%	30%
15	63	23.8%	41%	15	86	17.4%	41%	15	55	27.3%	41%	15	55	27.3%	41%
19	87	21.8%	51%	19	157	12.1%	51%	19	63	30.2%	51%	19	81	23.5%	51%
22	90	24.4%	59%	22	174	12.6%	59%	22	66	33.3%	59%	22	93	23.7%	59%
26	99	26.3%	70%	26	206	12.6%	70%	26	72	36.1%	70%	26	119	21.8%	70%
30	110	27.3%	81%	30	222	13.5%	81%	30	79	38.0%	81%	30	130	23.1%	81%
33	125	26.4%	89%	33	547	6.0%	89%	33	93	35.5%	89%	33	142	23.2%	89%
37	834	4.4%	100%	37	949	3.9%	100%	37	415	8.9%	100%	37	529	7.0%	100%

T416 *Optional* Queries

T416 Query 1											
X OR Y				X OR Y…n (9)				X OR Y…n (28)			
<keyword only> Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"				<hydro-electric.owl 9 S+S classes> from: Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"				<hydro-electric.owl 28 S+S+R classes> from: Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	1098			Retd	17230			Retd	42450		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	12	8.3%	10%	1	6	16.7%	10%	1	1	100.0%	10%
2	13	15.4%	20%	2	7	28.6%	20%	2	2	100.0%	20%
3	14	21.4%	30%	3	8	37.5%	30%	3	3	100.0%	30%
4	15	26.7%	40%	4	9	44.4%	40%	4	4	100.0%	40%
5	16	31.3%	50%	5	10	50.0%	50%	5	5	100.0%	50%
6	36	16.7%	60%	6	40	15.0%	60%	6	39	15.4%	60%
7	37	18.9%	70%	7	41	17.1%	70%	7	40	17.5%	70%
8	41	19.5%	80%	8	71	11.3%	80%	8	553	1.4%	80%
9	42	21.4%	90%	9	72	12.5%	90%	9	5010	0.2%	90%
10	956	1.0%	100%	10	329	3.0%	100%	10	5011	0.2%	100%

T416 Query 2											
X OR Y				X OR Y…n (7)				X OR Y…n (19)			
<keyword only> "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"				<hydro-electric.owl 7 S+S classes> from: "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"				<hydro-electric.owl 19 S+S+R classes> from: "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	8000			Retd	12242			Retd	38741		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	8	12.5%	10%	1	4	25.0%	10%	1	8	12.5%	10%
2	13	15.4%	20%	2	5	40.0%	20%	2	9	22.2%	20%
3	15	20.0%	30%	3	6	50.0%	30%	3	10	30.0%	30%
4	17	23.5%	40%	4	7	57.1%	40%	4	11	36.4%	40%
5	20	25.0%	50%	5	8	62.5%	50%	5	12	41.7%	50%
6	39	15.4%	60%	6	75	8.0%	60%	6	44	13.6%	60%
7	40	17.5%	70%	7	76	9.2%	70%	7	45	15.6%	70%
8	42	19.0%	80%	8	366	2.2%	80%	8	8035	0.1%	80%
9	43	20.9%	90%	9	367	2.5%	90%	9	12265	0.1%	90%
							0%	10	12266	0.1%	100%

T416 Query 3											
X OR Y				X OR Y…n (10)				X OR Y…n (29)			
<keyword only> Dam, "Hydro-electric Project", "total cost", "Three Gorges "				<hydro-electric.owl 10 S+S classes> from: Dam, "Hydro-electric Project", "total cost", "Three Gorges "				<hydro-electric.owl 29 S+S+R classes> from: Dam, "Hydro-electric Project", "total cost", "Three Gorges "			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	2100			Retd	17784			Retd	42784		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	10	10.0%	10%	1	14	7.1%	10%	1	1	100.0%	10%
2	11	18.2%	20%	2	15	13.3%	20%	2	2	100.0%	20%
3	12	25.0%	30%	3	16	18.8%	30%	3	3	100.0%	30%
4	13	30.8%	40%	4	17	23.5%	40%	4	4	100.0%	40%
5	14	35.7%	50%	5	18	27.8%	50%	5	5	100.0%	50%
6	47	12.8%	60%	6	30	20.0%	60%	6	21	28.6%	60%
7	48	14.6%	70%	7	31	22.6%	70%	7	22	31.8%	70%
8	49	16.3%	80%	8	32	25.0%	80%	8	732	1.1%	80%
9	50	18.0%	90%	9	33	27.3%	90%	9	3805	0.2%	90%
10	1963	0.5%	100%	10	518	1.9%	100%	10	3806	0.3%	100%

T416 Query 4											
X OR Y				X OR Y…n (6)				X OR Y…n (18)			
<keyword only> "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"				<hydro-electric.owl 6 S+S classes> from: "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"				<hydro-electric.owl 18 S+S+R classes> from: "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	46			Retd	5085			Retd	34706		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	1	100.0%	10%	1	1	100.0%	10%	1	5	20.0%	10%
2	2	100.0%	20%	2	2	100.0%	20%	2	6	33.3%	20%
3	3	100.0%	30%	3	3	100.0%	30%	3	7	42.9%	30%
4	4	100.0%	40%	4	4	100.0%	40%	4	8	50.0%	40%
5	5	100.0%	50%	5	5	100.0%	50%	5	9	55.6%	50%
6	6	100.0%	60%	6	24	25.0%	60%	6	183	3.3%	60%
7	7	100.0%	70%	7	25	28.0%	70%	7	184	3.8%	70%
8	14	57.1%	80%	8	78	10.3%	80%	8	2204	0.4%	80%
9	15	60.0%	90%	9	79	11.4%	90%	9	6415	0.1%	90%
								10	6416	0.2%	100%

T416 Query 5											
X OR Y				X OR Y…n (7)				X OR Y…n (19)			
<keyword only> "electrical output", "power station", "total cost", "three gorges dam"				<hydro-electric.owl 7 S+S classes> from: "electrical output", "power station", "total cost", "three gorges dam"				<hydro-electric.owl 19 S+S+R classes> from: "electrical output", "power station", "total cost", "three gorges dam"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	1491			Retd	5982			Retd	35130		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	2	50.0%	10%	1	2	50.0%	10%	1	7	14.3%	10%
2	3	66.7%	20%	2	3	66.7%	20%	2	8	25.0%	20%
3	4	75.0%	30%	3	4	75.0%	30%	3	9	33.3%	30%
4	5	80.0%	40%	4	5	80.0%	40%	4	10	40.0%	40%
5	6	83.3%	50%	5	6	83.3%	50%	5	11	45.5%	50%
6	16	37.5%	60%	6	24	25.0%	60%	6	795	0.8%	60%
7	17	41.2%	70%	7	25	28.0%	70%	7	796	0.9%	70%
8	57	14.0%	80%	8	81	9.9%	80%	8	2920	0.3%	80%
9	58	15.5%	90%	9	82	11.0%	90%	9	7030	0.1%	90%
								10	7031	0.1%	100%

T416 Query 6											
X OR Y				X OR Y…n (7)				X OR Y…n (21)			
<keyword only> "hydro-electric project", "total cost", reservoir, "three gorges"				<hydro-electric.owl 7 S+S classes> from: "hydro-electric project", "total cost", reservoir, "three gorges"				<hydro-electric.owl 21 S+S+R classes> from: "hydro-electric project", "total cost", reservoir, "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	2206			Retd	2553			Retd	33755		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	8	12.5%	7%	1	5	20.0%	13%	1	2	50.0%	33%
2	9	22.2%	14%	2	6	33.3%	25%	2	3	66.7%	67%
3	39	7.7%	21%	3	11	27.3%	38%	3	4	75.0%	100%
4	42	9.5%	29%	4	12	33.3%	50%	4	5	80.0%	133%
5	44	11.4%	36%	5	13	38.5%	63%	5	6	83.3%	167%
6	54	11.1%	43%	6	14	42.9%	75%	6	135	4.4%	200%
7	56	12.5%	50%	7	15	46.7%	88%	7	136	5.1%	233%
8	58	13.8%	57%	8	69	11.6%	100%	8	2697	0.3%	267%
9	61	14.8%	64%	9	71	12.7%	113%	9	2729	0.3%	300%
10	2084	0.5%	71%	10	2083	0.5%	125%	10	2730	0.4%	333%

T416 Query 7											
X OR Y				X OR Y…n (10)				X OR Y…n (29)			
<keyword only> dam, "completion date", "three gorges", "power station"				<hydro-electric.owl10 S+S classes> from: dam, "completion date", "three gorges", "power station"				<hydro-electric.owl 29 S+S+R classes> from: dam, "completion date", "three gorges", "power station"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Retd	10	Tot Doc	160838
Retd	1526			Retd	17267			Retd	42470		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	15	6.7%	10%	1	11	9.1%	10%	1	1	100.0%	10%
2	16	12.5%	20%	2	12	16.7%	20%	2	2	100.0%	20%
3	17	17.6%	30%	3	13	23.1%	30%	3	3	100.0%	30%
4	18	22.2%	40%	4	14	28.6%	40%	4	4	100.0%	40%
5	19	26.3%	50%	5	15	33.3%	50%	5	5	100.0%	50%
6	28	21.4%	60%	6	22	27.3%	60%	6	22	27.3%	60%
7	29	24.1%	70%	7	24	29.2%	70%	7	23	30.4%	70%
8	41	19.5%	80%	8	25	32.0%	80%	8	652	1.2%	80%
9	42	21.4%	90%	9	628	1.4%	90%	9	2837	0.3%	90%
10	1392	0.7%	100%	10	646	1.5%	100%	10	2838	0.4%	100%

T416 Query 8											
X OR Y				X OR Y…n (7)				X OR Y…n (19)			
<keyword only> "Yangtze River", "electrical output", "power station", "three gorges"				<hydro-electric.owl 7 S+S classes> from: "Yangtze River", "electrical output", "power station", "three gorges"				<hydro-electric.owl 19 S+S+R classes> from: "Yangtze River", "electrical output", "power station", "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	477			Retd	5100			Retd	34713		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	9	11.1%	10%	1	4	25.0%	10%	1	2	50.0%	10%
2	10	20.0%	20%	2	5	40.0%	20%	2	3	66.7%	20%
3	11	27.3%	30%	3	6	50.0%	30%	3	4	75.0%	30%
4	12	33.3%	40%	4	7	57.1%	40%	4	5	80.0%	40%
5	13	38.5%	50%	5	8	62.5%	50%	5	6	83.3%	50%
6	14	42.9%	60%	6	9	66.7%	60%	6	91	6.6%	60%
7	15	46.7%	70%	7	10	70.0%	70%	7	92	7.6%	70%
8	50	16.0%	80%	8	42	19.0%	80%	8	2185	0.4%	80%
9	53	17.0%	90%	9	45	20.0%	90%	9	2186	0.4%	90%
								10	2253	0.4%	100%

T416 Query 9											
X OR Y				X OR Y…n (8)				X OR Y…n (19)			
<keyword only> "electricity generation", "power station", "clean energy", "three gorges dam"				<hydro-electric.owl 8 S+S classes> from: "electricity generation", "power station", "clean energy", "three gorges dam"				<hydro-electric.owl 19 S+S+R classes> from: "electricity generation", "power station", "clean energy", "three gorges dam"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	707			Retd	712			Retd	32858		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	23	4.3%	10%	1	7	14.3%	10%	1	6	16.7%	10%
2	24	8.3%	20%	2	8	25.0%	20%	2	7	28.6%	20%
3	25	12.0%	30%	3	9	33.3%	30%	3	8	37.5%	30%
4	26	15.4%	40%	4	10	40.0%	40%	4	9	44.4%	40%
5	27	18.5%	50%	5	11	45.5%	50%	5	10	50.0%	50%
6	47	12.8%	60%	6	47	12.8%	60%	6	226	2.7%	60%
7	48	14.6%	70%	7	48	14.6%	70%	7	227	3.1%	70%
8	69	11.6%	80%	8	69	11.6%	80%	8	1877	0.4%	80%
9	70	12.9%	90%	9	70	12.9%	90%	9	1973	0.5%	90%
								10	1974	0.5%	100%

T416 Query 10											
X OR Y				X OR Y…n (6)				X OR Y…n (7)			
<keyword only> "flood control", "water storage", reservoir, "three gorges"				<hydro-electric.owl 6 S+S classes> from: "flood control", "water storage", reservoir, "three gorges"				<hydro-electric.owl 7 S+S+R classes> from: "flood control", "water storage", reservoir, "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	1364			Retd	1364			Retd	1574		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	29	3.4%	10%	1	29	3.4%	10%	1	28	3.6%	10%
2	30	6.7%	20%	2	30	6.7%	20%	2	29	6.9%	20%
3	31	9.7%	30%	3	31	9.7%	30%	3	31	9.7%	30%
4	32	12.5%	40%	4	32	12.5%	40%	4	32	12.5%	40%
5	33	15.2%	50%	5	33	15.2%	50%	5	33	15.2%	50%
6	34	17.6%	60%	6	34	17.6%	60%	6	34	17.6%	60%
7	35	20.0%	70%	7	35	20.0%	70%	7	35	20.0%	70%
8	47	17.0%	80%	8	47	17.0%	80%	8	47	17.0%	80%
9	52	17.3%	90%	9	52	17.3%	90%	9	52	17.3%	90%
10	1251	0.8%	100%	10	1251	0.8%	100%	10	1249	0.8%	100%

T416 *Must-have* Queries

T416 Query 1											
Must "Three Gorges Dam" X OR Y				Must "Three Gorges Dam" X OR Y…n (9)				Must "Three Gorges Dam" X OR Y…n (28)			
<keyword only> Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"				<hydro-electric.owl 9 S+S classes> from: Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"				<hydro-electric.owl 28 S+S+R classes> from: Dam, "Hydro-electric Project", "Yangtze River", "Three Gorges Dam"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	17 (1098)			Retd	17 (17230)			Retd	17 (42450)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	7	14.3%	10%	1	3	33.3%	10%	1	1	100.0%	10%
2	8	25.0%	20%	2	4	50.0%	20%	2	2	100.0%	20%
3	9	33.3%	30%	3	5	60.0%	30%	3	3	100.0%	30%
4	10	40.0%	40%	4	6	66.7%	40%	4	4	100.0%	40%
5	11	45.5%	50%	5	7	71.4%	50%	5	5	100.0%	50%
6	14	42.9%	60%	6	9	66.7%	60%	6	8	75.0%	60%
7	15	46.7%	70%	7	10	70.0%	70%	7	9	77.8%	70%
8	16	50.0%	80%	8	16	50.0%	80%	8	16	50.0%	80%
9	17	52.9%	90%	9	17	52.9%	90%	9	17	52.9%	90%

T416 Query 2											
Must "completion date" X OR Y				Must "completion date" X OR Y…n (7)				Must "completion date" X OR Y…n (19)			
<keyword only> "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"				<hydro-electric.owl 7 S+S classes> from: "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"				<hydro-electric.owl 19 S+S+R classes> from: "electrical output", "Three Gorges Dam", "Hydro-electric Project", "completion date"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	7967 (8000)			Retd	7967 (12242)			Retd	7967 (38741)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	6	16.7%	10%	1	46	2.2%	10%	1	29	3.4%	10%
2	7	28.6%	20%	2	47	4.3%	20%	2	30	6.7%	20%
3	9	33.3%	30%	3	50	6.0%	30%	3	33	9.1%	30%
4	10	40.0%	40%	4	52	7.7%	40%	4	35	11.4%	40%

T416 *Must-have* Queries (cont.)

T416 Query 3											
Must "Three Gorges " X OR Y				Must "Three Gorges " X OR Y…n (10)				Must "Three Gorges " X OR Y…n (29)			
<keyword only> Dam, "Hydro-electric Project", "total cost", "Three Gorges "				<hydro-electric.owl 10 S+S classes> from: Dam, "Hydro-electric Project", "total cost", "Three Gorges "				<hydro-electric.owl 29 S+S+R classes> from: Dam, "Hydro-electric Project", "total cost", "Three Gorges "			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	30 (2100)			Retd	30 (17784)			Retd	30 (42784)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	10	10.0%	10%	1	5	20.0%	10%	1	1	100.0%	10%
2	11	18.2%	20%	2	6	33.3%	20%	2	2	100.0%	20%
3	12	25.0%	30%	3	7	42.9%	30%	3	3	100.0%	30%
4	13	30.8%	40%	4	8	50.0%	40%	4	4	100.0%	40%
5	14	35.7%	50%	5	9	55.6%	50%	5	5	100.0%	50%
6	17	35.3%	60%	6	13	46.2%	60%	6	10	60.0%	60%
7	18	38.9%	70%	7	14	50.0%	70%	7	11	63.6%	70%
8	19	42.1%	80%	8	15	53.3%	80%	8	25	32.0%	80%
9	20	45.0%	90%	9	16	56.3%	90%	9	26	34.6%	90%

T416 Query 4											
Must "Three Gorges " X OR Y				Must "Three Gorges " X OR Y…n (6)				Must "Three Gorges " X OR Y…n (18)			
<keyword only> "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"				<hydro-electric.owl 6 S+S classes> from: "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"				<hydro-electric.owl 18 S+S+R classes> from: "electrical output", "hydro-electric project", "three gorges dam" , "three gorges project"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	10 (46)			Retd	10 (5085)			Retd	10 (34706)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	1	100.0%	10%	1	1	100.0%	10%	1	1	100.0%	10%
2	2	100.0%	20%	2	2	100.0%	20%	2	2	100.0%	20%
3	3	100.0%	30%	3	3	100.0%	30%	3	3	100.0%	30%
4	4	100.0%	40%	4	4	100.0%	40%	4	4	100.0%	40%
5	5	100.0%	50%	5	5	100.0%	50%	5	5	100.0%	50%
6	6	100.0%	60%	6	9	66.7%	60%	6	9	66.7%	60%
7	7	100.0%	70%	7	10	70.0%	70%	7	10	70.0%	70%

T416 Query 5											
Must "three gorges dam" X OR Y				Must "three gorges dam" X OR Y…n (7)				Must "three gorges dam" X OR Y…n (19)			
<keyword only> "electrical output", "power station", "total cost", "three gorges dam"				<hydro-electric.owl 7 S+S classes> from: "electrical output", "power station", "total cost", "three gorges dam"				<hydro-electric.owl 19 S+S+R classes> from: "electrical output", "power station", "total cost", "three gorges dam"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	17 (1491)			Retd	17 (5982)			Retd	17 (35130)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	1	100.0%	10%	1	1	100.0%	10%	1	3	33.3%	10%
2	2	100.0%	20%	2	2	100.0%	20%	2	4	50.0%	20%
3	3	100.0%	30%	3	3	100.0%	30%	3	5	60.0%	30%
4	4	100.0%	40%	4	4	100.0%	40%	4	6	66.7%	40%
5	5	100.0%	50%	5	5	100.0%	50%	5	7	71.4%	50%
6	6	100.0%	60%	6	6	100.0%	60%	6	9	66.7%	60%
7	7	100.0%	70%	7	7	100.0%	70%	7	10	70.0%	70%
8	16	50.0%	80%	8	16	50.0%	80%	8	16	50.0%	80%
9	17	52.9%	90%	9	17	52.9%	90%	9	17	52.9%	90%

T416 Query 6											
Must "Three Gorges " X OR Y				Must "Three Gorges " X OR Y…n (7)				Must "Three Gorges " X OR Y…n (21)			
<keyword only> "hydro-electric project", "total cost", reservoir, "three gorges"				<hydro-electric.owl 7 S+S classes> from: "hydro-electric project", "total cost", reservoir, "three gorges"				<hydro-electric.owl 21 S+S+R classes> from: "hydro-electric project", "total cost", reservoir, "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	30 (2206)			Retd	30 (2553)			Retd	30 (33755)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	5	20.0%	11%	1	5	20.0%	20%	1	1	100.0%	11%
2	6	33.3%	22%	2	6	33.3%	40%	2	2	100.0%	22%
3	8	37.5%	33%	3	9	33.3%	60%	3	3	100.0%	33%
4	11	36.4%	44%	4	10	40.0%	80%	4	4	100.0%	44%
5	13	38.5%	56%	5	11	45.5%	100%	5	5	100.0%	56%
6	23	26.1%	67%	6	12	50.0%	120%	6	13	46.2%	67%
7	25	28.0%	78%	7	13	53.8%	140%	7	14	50.0%	78%
8	27	29.6%	89%	8	17	47.1%	160%	8	25	32.0%	89%
9	30	30.0%	100%	9	19	47.4%	180%	9	26	34.6%	100%

T416 Query 7											
Must "power station" X OR Y				Must "power station" X OR Y…n (10)				Must "power station" X OR Y…n (29)			
<keyword only> dam, "completion date", "three gorges", "power station"				<hydro-electric.owl 10 S+S classes> from: dam, "completion date", "three gorges", "power station"				<hydro-electric.owl 29 S+S+R classes> from: dam, "completion date", "three gorges", "power station"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	425 (1526)			Retd	425 (17267)			Retd	425 (42470)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	8	12.5%	10%	1	7	14.3%	10%	1	1	100.0%	10%
2	9	22.2%	20%	2	8	25.0%	20%	2	2	100.0%	20%
3	10	30.0%	30%	3	9	33.3%	30%	3	3	100.0%	30%
4	11	36.4%	40%	4	10	40.0%	40%	4	4	100.0%	40%
5	12	41.7%	50%	5	11	45.5%	50%	5	5	100.0%	50%
6	13	46.2%	60%	6	15	40.0%	60%	6	233	2.6%	60%
7	14	50.0%	70%	7	16	43.8%	70%	7	234	3.0%	70%

T416 Query 8											
Must "three gorges" X OR Y				Must "three gorges" X OR Y…n (7)				Must "three gorges" X OR Y…n (19)			
<keyword only> "Yangtze River", "electrical output", "power station", "three gorges"				<hydro-electric.owl 7 S+S classes> from: "Yangtze River", "electrical output", "power station", "three gorges"				<hydro-electric.owl 19 S+S+R classes> from: "Yangtze River", "electrical output", "power station", "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	30 (477)			Retd	30 (5100)			Retd	17 (34713)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	8	12.5%	10%	1	4	25.0%	10%	1	2	50.0%	10%
2	9	22.2%	20%	2	5	40.0%	20%	2	3	66.7%	20%
3	10	30.0%	30%	3	6	50.0%	30%	3	4	75.0%	30%
4	11	36.4%	40%	4	7	57.1%	40%	4	5	80.0%	40%
5	12	41.7%	50%	5	8	62.5%	50%	5	6	83.3%	50%
6	13	46.2%	60%	6	9	66.7%	60%	6	13	46.2%	60%
7	14	50.0%	70%	7	10	70.0%	70%	7	14	50.0%	70%
8	16	50.0%	80%	8	16	50.0%	80%	8	27	29.6%	80%
9	19	47.4%	90%	9	19	47.4%	90%	9	28	32.1%	90%

T416 Query 9											
Must "three gorges dam" X OR Y				Must "three gorges dam" X OR Y…n (8)				Must "three gorges dam" X OR Y…n (19)			
<keyword only> "electricity generation", "power station", "clean energy", "three gorges dam"				<hydro-electric.owl 8 S+S classes> from: "electricity generation", "power station", "clean energy", "three gorges dam"				<hydro-electric.owl 19 S+S+R classes> from: "electricity generation", "power station", "clean energy", "three gorges dam""			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	17 (707)			Retd	17 (712)			Retd	17 (32858)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	1	100.0%	10%	1	1	100.0%	10%	1	3	33.3%	10%
2	2	100.0%	20%	2	2	100.0%	20%	2	4	50.0%	20%
3	3	100.0%	30%	3	3	100.0%	30%	3	5	60.0%	30%
4	4	100.0%	40%	4	4	100.0%	40%	4	6	66.7%	40%
5	5	100.0%	50%	5	5	100.0%	50%	5	7	71.4%	50%
6	6	100.0%	60%	6	6	100.0%	60%	6	9	66.7%	60%
7	7	100.0%	70%	7	7	100.0%	70%	7	10	70.0%	70%
8	16	50.0%	80%	8	16	50.0%	80%	8	16	50.0%	80%
9	17	52.9%	90%	9	17	52.9%	90%	9	17	52.9%	90%

T416 Query 10											
Must "three gorges" X OR Y				Must "three gorges" X OR Y…n (6)				Must "three gorges" X OR Y…n (7)			
<keyword only> "flood control", "water storage", reservoir, "three gorges"				<hydro-electric.owl 6 S+S classes> from: "flood control", "water storage", reservoir, "three gorges"				<hydro-electric.owl 7 S+S+R classes> from: "flood control", "water storage", reservoir, "three gorges"			
Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838	Relv Doc	10	Tot Doc	160838
Retd	30 (1364)			Retd	30 (1364)			Retd	17 (1574)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
1	3	33.3%	10%	1	3	33.3%	10%	1	3	33.3%	10%
2	4	50.0%	20%	2	4	50.0%	20%	2	4	50.0%	20%
3	5	60.0%	30%	3	5	60.0%	30%	3	5	60.0%	30%
4	6	66.7%	40%	4	6	66.7%	40%	4	6	66.7%	40%
5	7	71.4%	50%	5	7	71.4%	50%	5	7	71.4%	50%
6	8	75.0%	60%	6	8	75.0%	60%	6	8	75.0%	60%
7	9	77.8%	70%	7	9	77.8%	70%	7	9	77.8%	70%
8	11	72.7%	80%	8	11	72.7%	80%	8	11	72.7%	80%
9	16	56.3%	90%	9	16	56.3%	90%	9	16	56.3%	90%

T438 *Optional* and *Must-have* Queries

T438 Query 1

X OR Y			
<keyword only>"tourist destination", "package holiday", "foreign tourist", "tourism industry"			
Relv Doc	36	Tot Doc	96885
Retd	198	Folders	480
10% Rels	Rtd	P	R
4	18	22.2%	11%
7	30	23.3%	19%
11	61	18.0%	31%
14	78	17.9%	39%
18	134	13.4%	50%
20	173	11.6%	61%

X OR Y…n (14)			
<tourism-uk.owl 14 S+S classes> from:"tourist destination", "package holiday", "foreign tourist", "tourism industry"			
Relv Doc	36	Tot Doc	96885
Retd	2879		
10% Rels	Rtd	P	R
4	33	12.1%	11%
7	68	10.3%	19%
11	77	14.3%	31%
14	213	6.6%	39%
18	254	7.1%	50%
22	295	7.5%	61%
25	425	5.9%	69%
29	630	4.6%	81%
32	1476	2.2%	89%
33	2806	1.2%	100%

Must "tourism industry" X OR Y			
<keyword only>"tourist destination", "package holiday", "foreign tourist", "tourism industry"			
Relv Doc	36	Tot Doc	96885
Retd	133 (198)		
10% Rels	Rtd	P	R
4	12	33.3%	11%
7	45	15.6%	19%
11	108	10.2%	31%

Must "tourism industry" X OR Y…n (14)			
<tourism-uk.owl 14 S+S classes> from: "tourist destination", "package holiday", "foreign tourist", "tourism industry"			
Relv Doc	36	Tot Doc	96885
Retd	133 (2879)		
10% Rels	Rtd	P	R
4	35	11.4%	11%
7	39	17.9%	19%
11	124	8.9%	31%

T438 Query 2

X OR Y			
<keyword only> tourist, "tourist activity", holiday, "tour operator"			
Relv Doc	36	Tot Doc	96885
Retd	20879		
10% Rels	Rtd	P	R
4	149	2.7%	11%
7	214	3.3%	19%
11	316	3.5%	31%
14	329	4.3%	39%
18	397	4.5%	50%
22	417	5.3%	61%
25	548	4.6%	69%
29	856	3.4%	81%
32	1464	2.2%	89%

X OR Y…n (29)			
<tourism-uk.owl 29 S+S classes> from: tourist, "tourist activity", holiday, "tour operator"			
Relv Doc	36	Tot Doc	96885
Retd	20917		
10% Rels	Rtd	P	R
4	93	4.3%	11%
7	145	4.8%	19%
11	255	4.3%	31%
14	394	3.6%	39%
18	540	3.3%	50%
22	610	3.6%	61%
25	705	3.5%	69%
29	983	3.0%	81%
32	1567	2.0%	89%

Must "tour operator" X OR Y			
<keyword only> tourist, "tourist activity", holiday, "tour operator"			
Relv Doc	36	Tot Doc	96885
Retd	12784 (20879)		
10% Rels	Rtd	P	R
	0	0.0%	0%

Must "tour operator" X OR Y…n (29)			
<tourism-uk.owl 29 S+S classes> from: tourist, "tourist activity", holiday, "tour operator"			
Relv Doc	36	Tot Doc	96885
Retd	12784 (20917)		
10% Rels	Rtd	P	R
	0	0.0%	0%

T438 Query 3

X OR Y			
<keyword only>tourist, "tourist activity, increase, country			
Relv Doc	36	Tot Doc	96885
Retd	20879		
10% Rels	Rtd	P	R
4	73	5.5%	11%
7	165	4.2%	19%
11	327	3.4%	31%
14	377	3.7%	39%
18	383	4.7%	50%
22	827	2.7%	61%
25	2453	1.0%	69%
29	2866	1.0%	81%
32	3539	0.9%	89%
36	20413	0.2%	100%

X OR Y…n (21)			
<tourism-uk.owl 21 S+S classes> from: tourist, "tourist activity, increase, country			
Relv Doc	36	Tot Doc	96885
Retd	20917		
10% Rels	Rtd	P	R
4	94	4.3%	11%
7	159	4.4%	19%
11	297	3.7%	31%
14	359	3.9%	39%
18	752	2.4%	50%
22	838	2.6%	61%
25	2484	1.0%	69%
29	3240	0.9%	81%
32	4538	0.7%	89%
36	20451	0.2%	100%

Must "increase" X OR Y			
<keyword only>tourist, "tourist activity, increase, country			
Relv Doc	36	Tot Doc	96885
Retd	12784 (20879)		
10% Rels	Rtd	P	R
4	57	7.0%	11%
7	145	4.8%	19%
11	220	5.0%	31%
14	223	6.3%	39%
18	282	6.4%	50%
22	1907	1.2%	61%
25	2285	1.1%	69%
29	7669	0.4%	81%
32	12318	0.3%	89%

Must "increase" X OR Y…n (21)			
<tourism-uk.owl 21 S+S classes> from: tourist, "tourist activity, increase, country			
Relv Doc	36	Tot Doc	96885
Retd	12784 (20917)		
10% Rels	Rtd	P	R
4	61	6.6%	11%
7	125	5.6%	19%
11	189	5.8%	31%
14	222	6.3%	39%
18	288	6.3%	50%
22	1913	1.2%	61%
25	2289	1.1%	69%
29	7670	0.4%	81%
32	12318	0.3%	89%

T438 Query 4

X OR Y			
<keyword only>tourist, "tourist activity", "tourism industry", holiday			
Relv Doc	36	Tot Doc	96885
Retd	2805		
10% Rels	Rtd	P	R
4	28	14.3%	11%
7	180	3.9%	19%
11	234	4.7%	31%
14	246	5.7%	39%
18	350	5.1%	50%
22	372	5.9%	61%
25	440	5.7%	69%
29	858	3.4%	81%
32	1475	2.2%	89%

X OR Y…n (34)			
<tourism-uk.owl 34 S+S classes> from: tourist, "tourist activity", "tourism industry", holiday			
Relv Doc	36	Tot Doc	96885
Retd	3752		
10% Rels	Rtd	P	R
4	34	11.8%	11%
7	100	7.0%	19%
11	125	8.8%	31%
14	155	9.0%	39%
18	349	5.2%	50%
22	561	3.9%	61%
25	917	2.7%	69%
29	1055	2.7%	81%
32	2815	1.1%	89%

Must "Holiday" X OR Y			
<keyword only>tourist, "tourist activity", "tourism industry", holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (2805)		
10% Rels	Rtd	P	R
4	88	4.5%	11%
7	136	5.1%	19%
10	548	1.8%	31%

Must "Holiday" X OR Y…n (34)			
<tourism-uk.owl 34 S+S classes> from: tourist, "tourist activity", "tourism industry", holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (3752)		
10% Rels	Rtd	P	R
4	68	5.9%	11%
7	135	5.2%	19%
10	665	1.5%	31%

T438 Query 5

X OR Y			
<keyword only>"travel agent", "tourist activity", ecotourism, holiday			
Relv Doc	36	Tot Doc	96885
Retd	2114		
10% Rels	Rtd	P	R
4	528	0.8%	11%
7	1078	0.6%	19%
11	1998	0.6%	31%

X OR Y…n (20)			
<tourism-uk.owl 20 S+S classes> from: "travel agent", "tourist activity", ecotourism, holiday			
Relv Doc	36	Tot Doc	96885
Retd	3134		
10% Rels	Rtd	P	R
4	289	1.4%	11%
7	427	1.6%	19%
11	444	2.5%	31%
14	749	1.9%	39%
18	2035	0.9%	50%
19	2049	0.9%	61%

Must "Holiday" X OR Y			
<keyword only>"travel agent", "tourist activity", ecotourism, holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (2114)		
10% Rels	Rtd	P	R
4	477	0.8%	11%
7	1338	0.5%	19%
10	1762	0.6%	31%

Must "Holiday" X OR Y…n (20)			
<tourism-uk.owl 20 S+S classes> from: "travel agent", "tourist activity", ecotourism, holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (3134)		
10% Rels	Rtd	P	R
4	194	2.1%	11%
7	1356	0.5%	19%
10	1731	0.6%	31%

T438 Query 6

X OR Y			
<keyword only> vacation, "travel agent", "tour operator", " holiday destination"			
Relv Doc	36	Tot Doc	96885
Retd	1290		
10% Rels	Rtd	P	R
4	385	1.0%	11%
6	438	1.4%	19%

X OR Y…n (13)			
<tourism-uk.owl 13 S+S classes> from: vacation, "travel agent", "tour operator", " holiday destination"			
Relv Doc	36	Tot Doc	96885
Retd	5884		
10% Rels	Rtd	P	R
4	231	1.7%	11%
7	1320	0.5%	19%
11	1688	0.7%	31%
14	2969	0.5%	39%
18	3264	0.6%	50%
22	3532	0.6%	61%
25	3628	0.7%	69%
29	5832	0.5%	81%

Must "vacation" X OR Y			
<keyword only> vacation, "travel agent", "tour operator", " holiday destination"			
Relv Doc	36	Tot Doc	96885
Retd	1099 (1290)		
10% Rels	Rtd	P	R
4	247	1.6%	11%

Must "vacation" X OR Y…n (13)			
<tourism-uk.owl 13 S+S classes> from: vacation, "travel agent", "tour operator", " holiday destination"			
Relv Doc	36	Tot Doc	96885
Retd	1099 (5884)		
10% Rels	Rtd	P	R
4	1081	0.4%	11%

T438 Query 7

X OR Y			
<keyword only> increase, vacation, "tourism organisation", tourist			
Relv Doc	36	Tot Doc	96885
Retd	14072		
10% Rels	Rtd	P	R
4	292	1.4%	11%
7	296	2.4%	19%
11	416	2.6%	31%
14	452	3.1%	39%
18	609	3.0%	50%
22	1714	1.3%	61%
25	1880	1.3%	69%
29	1918	1.5%	81%
32	5251	0.6%	89%
36	13478	0.3%	100%

X OR Y…n (22)			
<tourism-uk.owl 22 S+S classes>from: increase, vacation, "tourism organisation", tourist			
Relv Doc	36	Tot Doc	96885
Retd	16373		
10% Rels	Rtd	P	R
4	249	1.6%	11%
7	463	1.5%	19%
11	534	2.1%	31%
14	621	2.3%	39%
18	707	2.5%	50%
22	2238	1.0%	61%
25	2899	0.9%	69%
29	3087	0.9%	81%
32	5733	0.6%	89%
36	15790	0.2%	100%

Must "tourist" X OR Y			
<keyword only> increase, vacation, "tourism organisation", tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (14072)		
10% Rels	Rtd	P	R
4	193	2.1%	11%
7	196	3.6%	19%
11	281	3.9%	31%
14	308	4.5%	39%
18	458	3.9%	50%
22	880	2.5%	61%
25	965	2.6%	69%
29	991	2.9%	81%
30	1011	3.0%	89%

Must "tourist" X OR Y…n (22)			
<tourism-uk.owl 22 S+S classes>from: increase, vacation, "tourism organisation", tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (16373)		
10% Rels	Rtd	P	R
4	165	2.4%	11%
7	284	2.5%	19%
11	328	3.4%	31%
14	375	3.7%	39%
18	427	4.2%	50%
22	887	2.5%	61%
25	966	2.6%	69%
29	986	2.9%	81%
30	999	3.0%	89%

T438 Query 8

X OR Y			
<keyword only> ecotourism, "foreign tourist", sightseeing, holiday			
Relv Doc	36	Tot Doc	96885
Retd	2078		
10% Rels	Rtd	P	R
4	40	10.0%	11%
7	46	15.2%	19%
11	241	4.6%	31%
14	667	2.1%	39%
18	1957	0.9%	50%

X OR Y…n (17)			
<tourism-uk.owl 17 S+S classes> from: ecotourism, "foreign tourist", sightseeing, holiday			
Relv Doc	36	Tot Doc	96885
Retd	3735		
10% Rels	Rtd	P	R
4	54	7.4%	11%
7	96	7.3%	19%
11	279	3.9%	31%
14	285	4.9%	39%
18	323	5.6%	50%
22	991	2.2%	61%
25	2284	1.1%	69%
29	2353	1.2%	81%
32	2751	1.2%	89%

Must "holiday" X OR Y			
<keyword only> ecotourism, "foreign tourist", sightseeing, holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (2078)		
10% Rels	Rtd	P	R
4	271	1.5%	11%
7	472	1.5%	19%
10	1757	0.6%	31%

Must "holiday" X OR Y…n (17)			
<tourism-uk.owl 17 S+S classes> from: ecotourism, "foreign tourist", sightseeing, holiday			
Relv Doc	36	Tot Doc	96885
Retd	1878 (3735)		
10% Rels	Rtd	P	R
4	74	5.4%	11%
7	247	2.8%	19%
10	1840	0.5%	31%

T438 Query 9

X OR Y			
<keyword only> "foreign country", "tourism industry", "beach resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2350		
10% Rels	Rtd	P	R
4	111	3.6%	11%
7	130	5.4%	19%
11	187	5.9%	31%
14	851	1.6%	39%
18	2882	0.6%	50%

X OR Y…n (11)			
<tourism-uk.owl 11 S+S classes> from: "foreign country", "tourism industry", "beach resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	3577		
10% Rels	Rtd	P	R
4	60	6.7%	11%
7	211	3.3%	19%
11	420	2.6%	31%
14	798	1.8%	39%
18	2281	0.8%	50%
22	2585	0.9%	61%
25	2675	0.9%	69%
29	3155	0.9%	81%
32	3481	0.9%	89%

Must "abroad" X OR Y			
<keyword only> "foreign country", "tourism industry", "beach resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2101 (2350)		
10% Rels	Rtd	P	R
4	438	0.9%	11%
7	2018	0.3%	19%
9	2033	0.4%	31%

Must "abroad" X OR Y…n (11)			
<tourism-uk.owl 11 S+S classes> from: "foreign country", "tourism industry", "beach resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2101 (3577)		
10% Rels	Rtd	P	R
4	190	2.1%	11%
7	2020	0.3%	19%
9	2033	0.4%	31%

T438 Query 10

X OR Y			
<keyword only> "ski resort", "beach resort", "package holiday", "travel agent"			
Relv Doc	36	Tot Doc	96885
Retd	281		
10% Rels	Rtd	P	R
1	229	0.4%	11%

X OR Y…n (7)			
<tourism-uk.owl 7 S+S classes> from: "ski resort", "beach resort", "package holiday", "travel agent"			
Relv Doc	36	Tot Doc	96885
Retd	3165		
10% Rels	Rtd	P	R
4	418	1.0%	11%
7	662	1.1%	19%
11	677	1.6%	31%
14	765	1.8%	39%
18	889	2.0%	50%
22	1875	1.2%	61%
25	3059	0.8%	69%

Must "travel agent" X OR Y			
<keyword only> "ski resort", "beach resort", "package holiday", "travel agent"			
Relv Doc	36	Tot Doc	96885
Retd	177 (281)		
10% Rels	Rtd	P	R
1	125	0.8%	11%

Must "travel agent" X OR Y…n (7)			
<tourism-uk.owl 7 S+S classes> from: "ski resort", "beach resort", "package holiday", "travel agent"			
Relv Doc	36	Tot Doc	96885
Retd	177 (3165)		
10% Rels	Rtd	P	R
1	151	0.7%	11%

T438 Query 11

X OR Y			
<keyword only> abroad, sightseeing, "tourist destination", tourist			
Relv Doc	36	Tot Doc	96885
Retd	3002		
10% Rels	Rtd	P	R
4	187	2.1%	11%
7	197	3.6%	19%
11	247	4.5%	31%
14	341	4.1%	39%
18	400	4.5%	50%
22	513	4.3%	61%
25	567	4.4%	69%
29	854	3.4%	81%
31	2943	1.1%	89%

X OR Y…n (16)			
<tourism-uk.owl 16 S+S classes> from: abroad, sightseeing, "tourist destination", tourist			
Relv Doc	36	Tot Doc	96885
Retd	3008		
10% Rels	Rtd	P	R
4	103	3.9%	11%
7	214	3.3%	19%
11	223	4.9%	31%
14	240	5.8%	39%
18	325	5.5%	50%
22	441	5.0%	61%
25	541	4.6%	69%
29	850	3.4%	81%
31	2944	1.1%	89%

Must "tourist" X OR Y			
<keyword only> abroad, sightseeing, "tourist destination", tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (3002)		
10% Rels	Rtd	P	R
4	142	2.8%	11%
7	152	4.6%	19%
11	202	5.4%	31%
14	294	4.8%	39%
18	353	5.1%	50%
22	466	4.7%	61%
25	520	4.8%	69%
29	807	3.6%	81%
30	809	3.7%	89%

Must "tourist" X OR Y…n (16)			
<tourism-uk.owl 16 S+S classes> from: abroad, sightseeing, "tourist destination", tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (3008)		
10% Rels	Rtd	P	R
4	101	4.0%	11%
7	168	4.2%	19%
11	177	6.2%	31%
14	194	7.2%	39%
18	277	6.5%	50%
22	393	5.6%	61%
25	493	5.1%	69%
29	802	3.6%	81%
30	814	3.7%	89%

T438 Query 12

X OR Y			
<keyword only> "tourism organisation", "holiday destination", "ski resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2161		
10% Rels	Rtd	P	R
4	479	0.8%	11%
7	2078	0.3%	19%
9	2093	0.4%	31%

X OR Y…n (xx)			
<tourism-uk.owl xx S+S classes> from: "tourism organisation", "holiday destination", "ski resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	5465		
10% Rels	Rtd	P	R
4	101	4.0%	11%
7	228	3.1%	19%
11	1020	1.1%	31%
14	2159	0.6%	39%
18	2502	0.7%	50%
22	2689	0.8%	61%
25	3011	0.8%	69%
29	4549	0.6%	81%

Must "abroad" X OR Y			
<keyword only> "tourism organisation", "holiday destination", "ski resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2101 (2161)		
10% Rels	Rtd	P	R
4	570	0.7%	11%
7	2018	0.3%	19%
9	2033	0.4%	31%

Must "abroad" X OR Y…n (xx)			
<tourism-uk.owl xx S+S classes> from: "tourism organisation", "holiday destination", "ski resort", abroad			
Relv Doc	36	Tot Doc	96885
Retd	2101 (5465)		
10% Rels	Rtd	P	R
4	210	1.9%	11%
7	1981	0.4%	19%
9	1994	0.5%	31%

T438 Query 13															
X OR Y				X OR Y…n (17)				Must "resort" X OR Y				Must "resort" X OR Y…n (17)			
<keyword only> "foreign tourist", tourist, vacation, resort				<tourism-uk.owl 16 S+S classes> from: "foreign tourist", tourist, vacation, resort				<keyword only> "foreign tourist", tourist, vacation, resort				<tourism-uk.owl 16 S+S classes> from: "foreign tourist", tourist, vacation, resort			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	3074			Retd	4537			Retd	1376 (3074)			Retd	1376 (4537)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	36	11.1%	11%	4	71	5.6%	11%	4	151	2.6%	11%	4	168	2.4%	11%
7	174	4.0%	19%	7	119	5.9%	19%	7	228	3.1%	19%	7	253	2.8%	19%
11	210	5.2%	31%	11	261	4.2%	31%	11	264	4.2%	31%	11	341	3.2%	31%
14	251	5.6%	39%	14	345	4.1%	39%	13	287	4.5%	39%	13	368	3.5%	39%
18	315	5.7%	50%	18	432	4.2%	50%								
22	364	6.0%	61%	22	538	4.1%	61%								
25	689	3.6%	69%	25	620	4.0%	69%								
29	991	2.9%	81%	29	986	2.9%	81%								
30	1043	2.9%	89%	32	3408	0.9%	89%								

T438 Query 14															
X OR Y				X OR Y…n (5)				Must "increase" X OR Y				Must "increase" X OR Y…n (5)			
<keyword only> country, tourism, vacation, increase				<tourism-uk.owl 16 S+S classes> from: country, tourism, vacation, increase				<keyword only> country, tourism, vacation, increase				<tourism-uk.owl 16 S+S classes> from: country, tourism, vacation, increase			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	21701			Retd	22549			Retd	12784 (21701)			Retd	12784 (22549)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	385	1.0%	11%	4	390	1.0%	11%	4	201	2.0%	11%	4	219	1.8%	11%
7	432	1.6%	19%	7	460	1.5%	19%	7	229	3.1%	19%	7	232	3.0%	19%
11	534	2.1%	31%	11	593	1.9%	31%	11	283	3.9%	31%	11	336	3.3%	31%
14	630	2.2%	39%	14	723	1.9%	39%	14	344	4.1%	39%	14	387	3.6%	39%
18	1240	1.5%	50%	18	1541	1.2%	50%	18	425	4.2%	50%	18	608	3.0%	50%
22	1253	1.8%	61%	22	1575	1.4%	61%	22	1717	1.3%	61%	22	1869	1.2%	61%
25	3246	0.8%	69%	25	3517	0.7%	69%	25	1744	1.4%	69%	25	1907	1.3%	69%
29	3278	0.9%	81%	29	3564	0.8%	81%	29	2196	1.3%	81%	29	2358	1.2%	81%
32	3678	0.9%	89%	32	4132	0.8%	89%	32	3670	0.9%	89%	32	3635	0.9%	89%
36	5286	0.7%	100%	36	8385	0.4%	100%								

T438 Query 15															
X OR Y				X OR Y…n (16)				Must "tourist" X OR Y				Must "tourist" X OR Y…n (16)			
<keyword only> "foreign tourist", increase, resort, tourist				<tourism-uk.owl 16 S+S classes> from: "foreign tourist", increase, resort, tourist				<keyword only> "foreign tourist", increase, resort, tourist				<tourism-uk.owl 16 S+S classes> from: "foreign tourist", increase, resort, tourist			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	14073			Retd	14077			Retd	1034 (14073)			Retd	1034 (14077)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	58	6.9%	11%	4	70	5.7%	11%	4	58	6.9%	11%	4	70	5.7%	11%
7	70	10.0%	19%	7	90	7.8%	19%	7	70	10.0%	19%	7	80	8.8%	19%
11	94	11.7%	31%	11	96	11.5%	31%	11	94	11.7%	31%	11	86	12.8%	31%
14	228	6.1%	39%	14	296	4.7%	39%	14	228	6.1%	39%	14	246	5.7%	39%
18	383	4.7%	50%	18	438	4.1%	50%	18	248	7.3%	50%	18	261	6.9%	50%
22	510	4.3%	61%	22	566	3.9%	61%	22	343	6.4%	61%	22	350	6.3%	61%
25	552	4.5%	69%	25	608	4.1%	69%	25	363	6.9%	69%	25	370	6.8%	69%
29	2037	1.4%	81%	29	1898	1.5%	81%	29	989	2.9%	81%	29	941	3.1%	81%
32	5305	0.6%	89%	32	5311	0.6%	89%	30	1011	3.0%	89%	30	1011	3.0%	89%
36	13477	0.3%	100%	36	13483	0.3%	100%								

T438 Query 16															
X OR Y				X OR Y…n (5)				Must "tourism" X OR Y				Must "tourism" X OR Y…n (5)			
<keyword only> country, ecotourism, increase, tourism				<tourism-uk.owl 16 S+S classes> from: country, ecotourism, increase, tourism				<keyword only> country, ecotourism, increase, tourism				<tourism-uk.owl 16 S+S classes> from: country, ecotourism, increase, tourism			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	21224			Retd	21224			Retd	1399 (21224)			Retd	1399 (21224)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	258	1.6%	11%	4	280	1.4%	11%	4	231	1.7%	11%	4	253	1.6%	11%
7	286	2.4%	19%	7	298	2.3%	19%	7	258	2.7%	19%	7	270	2.6%	19%
11	345	3.2%	31%	11	353	3.1%	31%	11	316	3.5%	31%	11	324	3.4%	31%
14	402	3.5%	39%	14	454	3.1%	39%	14	371	3.8%	39%	14	423	3.3%	39%
18	488	3.7%	50%	18	483	3.7%	50%	18	457	3.9%	50%	18	445	4.0%	50%
22	498	4.4%	61%	22	555	4.0%	61%	22	467	4.7%	61%	22	484	4.5%	61%
25	2462	1.0%	69%	25	2466	1.0%	69%	25	1225	2.0%	69%	25	1229	2.0%	69%
29	2688	1.1%	81%	29	2691	1.1%	81%	29	1304	2.2%	81%	29	1308	2.2%	81%
32	2961	1.1%	89%	32	2963	1.1%	89%	32	1331	2.4%	89%	32	1333	2.4%	89%
35	4481	0.8%	100%	35	4481	0.8%	100%	34	1376	2.5%	100%	34	1376	2.5%	100%

T438 Query 17															
X OR Y				X OR Y…n (15)				Must "resort" X OR Y				Must "resort" X OR Y…n (15)			
<keyword only> holiday, tourism, "tourist destination", resort				<tourism-uk.owl 15 S+S classes> from: holiday, tourism, "tourist destination", resort				<keyword only> holiday, tourism, "tourist destination", resort				<tourism-uk.owl 15 S+S classes> from: holiday, tourism, "tourist destination", resort			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	4173			Retd	4949			Retd	1376 (4173)			Retd	1376 (4949)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	34	11.8%	11%	4	64	6.3%	11%	4	31	12.9%	11%	4	58	6.9%	11%
7	198	3.5%	19%	7	211	3.3%	19%	7	225	3.1%	19%	7	261	2.7%	19%
11	243	4.5%	31%	11	290	3.8%	31%	11	265	4.2%	31%	11	353	3.1%	31%
14	308	4.5%	39%	14	330	4.2%	39%	14	302	4.6%	39%	14	379	3.7%	39%
18	335	5.4%	50%	18	449	4.0%	50%	18	321	5.6%	50%	18	420	4.3%	50%
22	385	5.7%	61%	22	541	4.1%	61%								
25	420	6.0%	69%	25	593	4.2%	69%								
29	1620	1.8%	81%	29	1721	1.7%	81%								
32	1766	1.8%	89%	32	1863	1.7%	89%								
36	2523	1.4%	100%	36	2581	1.4%	100%								

T438 Query 18															
X OR Y				X OR Y…n (18)				Must "increase" X OR Y				Must "increase" X OR Y…n (18)			
<keyword only> "foreign country", holiday, "tourism industry", increase				<tourism-uk.owl 18 S+S classes> from: "foreign country", holiday, "tourism industry", increase				<keyword only> "foreign country", holiday, "tourism industry", increase				<tourism-uk.owl 18 S+S classes> from: "foreign country", holiday, "tourism industry", increase			
Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885	Relv Doc	36	Tot Doc	96885
Retd	14200			Retd	14913			Retd	12784 (14200)			Retd	12784 (14913)		
10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R	10% Rels	Rtd	P	R
4	77	5.2%	11%	4	94	4.3%	11%	4	227	1.8%	11%	4	231	1.7%	11%
7	372	1.9%	19%	7	380	1.8%	19%	7	456	1.5%	19%	7	398	1.8%	19%
11	1872	0.6%	31%	11	633	1.7%	31%	11	652	1.7%	31%	11	669	1.6%	31%
14	1968	0.7%	39%	14	2066	0.7%	39%	14	691	2.0%	39%	14	855	1.6%	39%
18	2107	0.9%	50%	18	2280	0.8%	50%	18	2985	0.6%	50%	18	3204	0.6%	50%
22	4401	0.5%	61%	22	4656	0.5%	61%	22	3867	0.6%	61%	22	4066	0.5%	61%
25	5280	0.5%	69%	25	4687	0.5%	69%	25	3890	0.6%	69%	25	4086	0.6%	69%
29	5306	0.5%	81%	29	5538	0.5%	81%	29	6571	0.4%	81%	29	6731	0.4%	81%
32	7985	0.4%	89%	32	7636	0.4%	89%	32	12160	0.3%	89%	32	12170	0.3%	89%
36	13576	0.3%	100%	36	13622	0.3%	100%								

T438 Query 19

X OR Y			
<keyword only> "foreign tourist", "holiday destination", tourism, tourist			
Relv Doc	36	Tot Doc	96885
Retd	2039		
10% Rels	Rtd	P	R
4	26	15.4%	11%
7	49	14.3%	19%
11	174	6.3%	31%
14	187	7.5%	39%
18	231	7.8%	50%
22	283	7.8%	61%
25	394	6.3%	69%
29	541	5.4%	81%
32	1202	2.7%	89%
36	1984	1.8%	100%

X OR Y…n (21)			
<tourism-uk.owl 21 S+S classes> from: "foreign tourist", "holiday destination", tourism, tourist			
Relv Doc	36	Tot Doc	96885
Retd	5015		
10% Rels	Rtd	P	R
4	35	11.4%	11%
7	75	9.3%	19%
11	172	6.4%	31%
14	187	7.5%	39%
18	222	8.1%	50%
22	383	5.7%	61%
25	490	5.1%	69%
29	1069	2.7%	81%
32	1261	2.5%	89%
36	2081	1.7%	100%

Must "tourist" X OR Y			
<keyword only> "foreign tourist", "holiday destination", tourism, tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (2039)		
10% Rels	Rtd	P	R
4	25	16.0%	11%
7	48	14.6%	19%
11	173	6.4%	31%
14	186	7.5%	39%
18	230	7.8%	50%
22	282	7.8%	61%
25	393	6.4%	69%
29	540	5.4%	81%
30	1028	2.9%	89%

Must "tourist" X OR Y…n (21)			
<tourism-uk.owl 21 S+S classes> from: "foreign tourist", "holiday destination", tourism, tourist			
Relv Doc	36	Tot Doc	96885
Retd	1034 (5015)		
10% Rels	Rtd	P	R
4	35	11.4%	11%
7	74	9.5%	19%
11	171	6.4%	31%
14	182	7.7%	39%
18	217	8.3%	50%
22	276	8.0%	61%
25	477	5.2%	69%
29	1019	2.8%	81%
30	1020	2.9%	89%

T438 Query 20

X OR Y			
<keyword only> "beach resort", resort, "tourist destination", tourism			
Relv Doc	36	Tot Doc	96885
Retd	2593		
10% Rels	Rtd	P	R
4	52	7.7%	11%
7	185	3.8%	19%
11	194	5.7%	31%
14	216	6.5%	39%
18	239	7.5%	50%
22	1495	1.5%	61%
25	1591	1.6%	69%
29	1753	1.7%	81%
32	1792	1.8%	89%
36	2515	1.4%	100%

X OR Y…n (6)			
<tourism-uk.owl 6 S+S classes> from: "beach resort", resort, "tourist destination", tourism			
Relv Doc	36	Tot Doc	96885
Retd	2595		
10% Rels	Rtd	P	R
4	60	6.7%	11%
7	220	3.2%	19%
11	240	4.6%	31%
14	246	5.7%	39%
18	284	6.3%	50%
22	1493	1.5%	61%
25	1539	1.6%	69%
29	1695	1.7%	81%
32	1794	1.8%	89%
36	2517	1.4%	100%

Must "tourism" X OR Y			
<keyword only> "beach resort", resort, "tourist destination", tourism			
Relv Doc	36	Tot Doc	96885
Retd	1399 (2593)		
10% Rels	Rtd	P	R
4	33	12.1%	11%
7	145	4.8%	19%
11	154	7.1%	31%
14	175	8.0%	39%
18	194	9.3%	50%
22	341	6.5%	61%
25	485	5.2%	69%
29	597	4.9%	81%
32	1309	2.4%	89%
34	1321	2.6%	100%

Must "tourism" X OR Y…n (6)			
<tourism-uk.owl 6 S+S classes> from: "beach resort", resort, "tourist destination", tourism			
Relv Doc	36	Tot Doc	96885
Retd	1399 (2595)		
10% Rels	Rtd	P	R
4	39	10.3%	11%
7	146	4.8%	19%
11	156	7.1%	31%
14	162	8.6%	39%
18	193	9.3%	50%
22	339	6.5%	61%
25	403	6.2%	69%
29	597	4.9%	81%
32	1309	2.4%	89%
34	1321	2.6%	100%

T401 vs. T401+SUMO Q1-10 Group Average %s Query Data.

Oo T401 vs. T401+SUMO - MEA-based %			
T401		T401+SUMO	
P	R	P	R
39.8%	10%	39.3%	10%
41.1%	20%	40.0%	20%
39.3%	30%	36.2%	30%
35.5%	40%	32.2%	40%
34.1%	50%	29.0%	50%
32.9%	60%	28.4%	60%
28.8%	70%	25.3%	70%
25.4%	80%	19.8%	80%
21.9%	90%	15.1%	90%
10.4%	100%	6.6%	100%

Oo T401 vs. T401+SUMO (Unit %)			
T401		T401+SUMO	
P	R	P	R
10.2%	10%	10.3%	10%
14.1%	20%	12.1%	20%
16.6%	30%	12.5%	30%
19.0%	40%	12.7%	40%
18.1%	50%	11.1%	50%
18.8%	60%	10.8%	60%
16.3%	70%	10.5%	70%
12.4%	80%	9.4%	80%
10.2%	90%	7.9%	90%
6.2%	100%	4.3%	100%

T401 vs. T401+SUMO Individual Query Data: Queries 1 to 10.

Q1 Oo T401				Query 1	Q1 Oo T401+SUMO			
X OR Y…n (7)					X OR Y…n (20)			
<immigration2.owl 7 S+S classes> from: "foreign minority", Germany, culture, integration					<immigration2.owl 20 S+S classes> from: "foreign minority", Germany, culture, integration			
Relv Doc	37	Tot Doc	13065		Relv Doc	37	Tot Doc	13065
Retd	1448				Retd	5215		
10% Rels	Rtd	P	R		10% Rels	Rtd	P	R
4	153	2.6%	11%		4	139	2.9%	11%
7	165	4.2%	19%		7	165	4.2%	19%
11	179	6.1%	30%		11	223	4.9%	30%
15	192	7.8%	41%		15	309	4.9%	41%
19	215	8.8%	51%		19	374	5.1%	51%
22	221	10.0%	59%		22	439	5.0%	59%
26	229	11.4%	70%		26	485	5.4%	70%
30	525	5.7%	81%		30	505	5.9%	81%
33	529	6.2%	89%		33	573	5.8%	89%
37	805	4.6%	100%		37	760	4.9%	100%

Q2 Oo T401				Query 2	Q2 Oo T401+SUMO			
X OR Y…n (7)					X OR Y…n (18)			
<immigration2.owl 7 S+S classes> from: "ethnic minority", "cultural difference", "immigration issue", Germany					<immigration2.owl 18 S+S classes> from: "ethnic minority", "cultural difference", "immigration issue", Germany			
Relv Doc	37	Tot Doc	13065		Relv Doc	37	Tot Doc	13065
Retd	917				Retd	4513		
10% Rels	Rtd	P	R		10% Rels	Rtd	P	R
4	102	3.9%	11%		4	86	4.7%	11%
7	112	6.3%	19%		7	144	4.9%	19%
11	125	8.8%	30%		11	207	5.3%	30%
15	134	11.2%	41%		15	269	5.6%	41%
19	143	13.3%	51%		19	282	6.7%	51%
22	151	14.6%	59%		22	325	6.8%	59%
26	158	16.5%	70%		26	345	7.5%	70%
30	499	6.0%	81%		30	379	7.9%	81%
33	524	6.3%	89%		33	393	8.4%	89%
37	600	6.2%	100%		37	580	6.4%	100%

T401 vs. T401+SUMO Individual Query Data

					Query 3	Q3 Oo T401+SUMO				
X OR Y…n (12)						X OR Y…n (34)				
<immigration2.owl 12 S+S classes> from: migrant, "cultural integration", protection, Germany						<immigration2.owl 34 S+S classes> from: migrant, "cultural integration", protection, Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	2883					Retd	8870			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	31	12.9%		11%		4	49	8.2%		11%
7	50	14.0%		19%		7	66	10.6%		19%
11	64	17.2%		30%		11	96	11.5%		30%
15	100	15.0%		41%		15	142	10.6%		41%
19	176	10.8%		51%		19	288	6.6%		51%
22	222	9.9%		59%		22	382	5.8%		59%
26	354	7.3%		70%		26	492	5.3%		70%
30	427	7.0%		81%		30	712	4.2%		81%
33	784	4.2%		89%		33	1062	3.1%		89%
37	905	4.1%		100%		37	1686	2.2%		100%

Q4 Oo T401					Query 4	Q4 Oo T401+SUMO				
X OR Y…n (7)						X OR Y…n (23)				
<immigration2.owl 7 S+S classes> from: "foreign minority", immigration, refugee, Germany						<immigration2.owl 23 S+S classes> from: "foreign minority", immigration, refugee, Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	886					Retd	7633			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	4	100.0%		11%		4	4	100.0%		11%
7	8	87.5%		19%		7	8	87.5%		19%
11	14	78.6%		30%		11	14	78.6%		30%
15	22	68.2%		41%		15	19	78.9%		41%
19	29	65.5%		51%		19	26	73.1%		51%
22	35	62.9%		59%		22	32	68.8%		59%
26	50	52.0%		70%		26	49	53.1%		70%
30	70	42.9%		81%		30	75	40.0%		81%
33	91	36.3%		89%		33	86	38.4%		89%
37	120	30.8%		100%		37	151	24.5%		100%

Q5 Oo T401					Query 5	Q5 Oo T401+SUMO				
X OR Y…n (6)						X OR Y…n (28)				
<immigration2.owl 6 S+S classes> from: "asylum seeker", employment, "foreign national", Germany						<immigration2.owl 28 S+S classes> from: "asylum seeker", employment, "foreign national", Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	1218					Retd	8838			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	8	50.0%		11%		4	6	66.7%		11%
7	11	63.6%		19%		7	12	58.3%		19%
11	16	68.8%		30%		11	16	68.8%		30%
15	24	62.5%		41%		15	31	48.4%		41%
19	30	63.3%		51%		19	40	47.5%		51%
22	35	62.9%		59%		22	46	47.8%		59%
26	55	47.3%		70%		26	68	38.2%		70%
30	67	44.8%		81%		30	101	29.7%		81%
33	73	45.2%		89%		33	155	21.3%		89%
37	206	18.0%		100%		37	657	5.6%		100%

T401 vs. T401+SUMO Individual Query Data

Q6 Oo T401					Query 6	Q6 Oo T401+SUMO				
X OR Y…n (6)						X OR Y…n (15)				
<immigration2.owl 6 S+S classes> from: migration, "immigration issue", culture, Germany						<immigration2.owl 15 S+S classes> from: migration, "immigration issue", culture, Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	1425					Retd	4731			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	43	9.3%		11%		4	35	11.4%		11%
7	67	10.4%		19%		7	82	8.5%		19%
11	130	8.5%		30%		11	162	6.8%		30%
15	143	10.5%		41%		15	179	8.4%		41%
19	177	10.7%		51%		19	225	8.4%		51%
22	181	12.2%		59%		22	276	8.0%		59%
26	360	7.2%		70%		26	363	7.2%		70%
30	373	8.0%		81%		30	402	7.5%		81%
33	387	8.5%		89%		33	578	5.7%		89%
37	1333	2.8%		100%		37	1039	3.6%		100%

Q7 Oo T401					Query 7	Q7 Oo T401+SUMO				
X OR Y…n (8)						X OR Y…n (33)				
<immigration2.owl 8 Sub+Sup classes> from: asylum, immigrant, "quality of life", Germany						<immigration2.owl 33 S+S classes> from: asylum, immigrant, "quality of life", Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	987		No doc list			Retd	8688			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	8	50.0%		11%		4	6	66.7%		11%
7	13	53.8%		19%		7	10	70.0%		19%
11	24	45.8%		30%		11	24	45.8%		30%
15	34	44.1%		41%		15	40	37.5%		41%
19	62	30.6%		51%		19	63	30.2%		51%
22	72	30.6%		59%		22	67	32.8%		59%
26	79	32.9%		70%		26	85	30.6%		70%
30	105	28.6%		81%		30	126	23.8%		81%
33	156	21.2%		89%		33	224	14.7%		89%
37	488	7.6%		100%		37	1013	3.7%		100%

Q8 Oo T401					Query 8	Q8 Oo T401+SUMO				
X OR Y…n (10)						Must "Germany" X OR Y…n (31)				
<immigration2.owl 10 Sub+Sup classes> from: "asylum seeker", security, shelter, Germany						<immigration2.owl 31 S+S classes> from: "asylum seeker", security, shelter, Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	2676		No doc list			Retd	8832			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	4	100.0%		11%		4	5	80.0%		11%
7	7	100.0%		19%		7	8	87.5%		19%
11	13	84.6%		30%		11	16	68.8%		30%
15	21	71.4%		41%		15	25	60.0%		41%
19	25	76.0%		51%		19	32	59.4%		51%
22	36	61.1%		59%		22	41	53.7%		59%
26	45	57.8%		70%		26	47	55.3%		70%
30	56	53.6%		81%		30	90	33.3%		81%
33	70	47.1%		89%		33	238	13.9%		89%
37	243	15.2%		100%		37	542	6.8%		100%

T401 vs. T401+SUMO Individual Query Data

Q9 Oo T401					Query 9	Q9 Oo T401+SUMO				
X OR Y…n (5)						Must "Germany" X OR Y…n (16)				
<immigration2.owl 5 Sub+Sup classes> from: "economic migrant", "illegal immigrant", "immigration control", Germany						<immigration2.owl 16 S+S classes> from: "economic migrant", "illegal immigrant", "immigration control", Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	612		No doc list			Retd	6820			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	7	57.1%		11%		4	9	44.4%		11%
7	12	58.3%		19%		7	12	58.3%		19%
11	18	61.1%		30%		11	18	61.1%		30%
15	32	46.9%		41%		15	26	57.7%		41%
19	38	50.0%		51%		19	40	47.5%		51%
22	42	52.4%		59%		22	44	50.0%		59%
26	61	42.6%		70%		26	57	45.6%		70%
30	69	43.5%		81%		30	73	41.1%		81%
33	86	38.4%		89%		33	94	35.1%		89%
37	342	10.8%		100%		37	641	5.8%		100%

Q10 Oo T401					Query 10	Q10 Oo T401+SUMO				
X OR Y…n (13)						Must "Germany" X OR Y…n (31)				
<immigration2.owl 13 Sub+Sup classes> from: "cultural difference", integration, migrant, Germany						<immigration2.owl 31 S+S classes> from: "cultural difference", integration, migrant, Germany				
Relv Doc	37	Tot Doc		13065		Relv Doc	37	Tot Doc		13065
Retd	1092		No doc list			Retd	7325			
10% Rels	Rtd	P		R		10% Rels	Rtd	P		R
4	34	11.8%		11%		4	50	8.0%		11%
7	53	13.2%		19%		7	71	9.9%		19%
11	80	13.8%		30%		11	104	10.6%		30%
15	86	17.4%		41%		15	143	10.5%		41%
19	157	12.1%		51%		19	339	5.6%		51%
22	174	12.6%		59%		22	390	5.6%		59%
26	206	12.6%		70%		26	494	5.3%		70%
30	222	13.5%		81%		30	713	4.2%		81%
33	547	6.0%		89%		33	776	4.3%		89%
37	949	3.9%		100%		37	1521	2.4%		100%

APPENDIX G: OTHER TOPIC PRECISION & RECALL GRAPHS

This Appendix section contains those P&R graphs not shown in the main experiment results sections. The graphs are organised as follows.

1. T401 *Optional* Mode P&R Graphs – pages LIX and LX.

    Fig. A17. T401 Q1 *optional* mode P&R.

    Fig. A18. T401 Q2 *optional* mode P&R.

    Fig. A19. T401 Q3 *optional* mode P&R.

    Fig. A20. T401 Q5 *optional* mode P&R.

2. T401 *Must-have* Mode P&R Graphs – pages LX to LXIII.

    Fig. A21. T401 Q1 *must-have* mode P&R.

    Fig. A22. T401 Q2 *must-have* mode P&R.

    Fig. A23. T401 Q3 *must-have* mode P&R.

    Fig. A24. T401 Q4 *must-have* mode P&R.

    Fig. A25. T401 Q5 *must-have* mode P&R.

    Fig. A26. T401 Q6 *must-have* mode P&R.

    Fig. A27. T401 Q7 *must-have* mode P&R.

    Fig. A28. T401 Q8 *must-have* mode P&R.

    Fig. A29. T401 Q9 *must-have* mode P&R.

    Fig. A30. T401 Q10 *must-have* mode P&R.

3. T416 *Optional* Mode P&R Graphs – pages LXIII to LXV.

    Fig. A31. T416 Q2 *optional* mode P&R.

    Fig. A32. T416 Q3 *optional* mode P&R.

    Fig. A33. T416 Q4 *optional* mode P&R.

    Fig. A34. T416 Q6 *optional* mode P&R.

    Fig. A35. T416 Q7 *optional* mode P&R.

    Fig. A36. T416 Q9 *optional* mode P&R.

4. T416 *Must-have* Mode P&R Graphs - pages LXV to LXVII.

    Fig. A37. T416 Q2 *must-have* mode P&R.

    Fig. A38. T416 Q3 *must-have* mode P&R.

    Fig. A39. T416 Q4 *must-have* mode P&R.

    Fig. A40. T416 Q6 *must-have* mode P&R.

    Fig. A41. T416 Q7 *must-have* mode P&R.

    Fig. A42. T416 Q9 *must-have* mode P&R.

T401 *Optional* Mode P&R Graphs (not shown in main body).

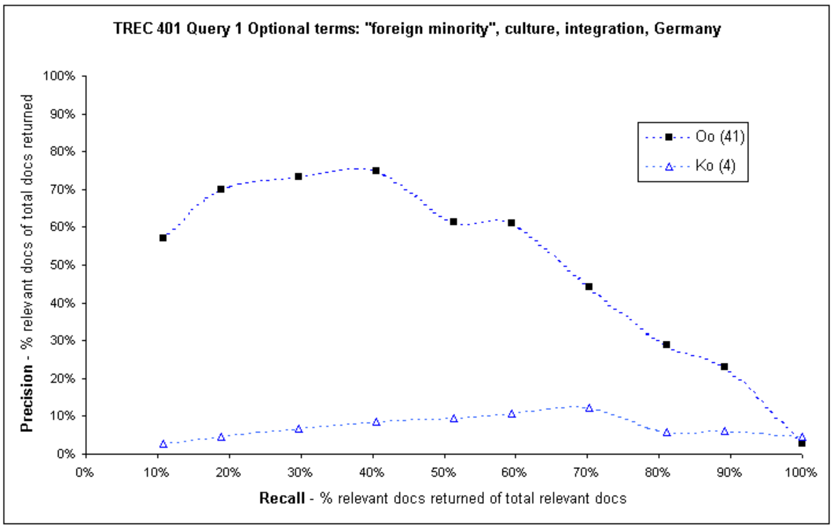


Fig. A17. T401 Q1 *optional* mode P&R.

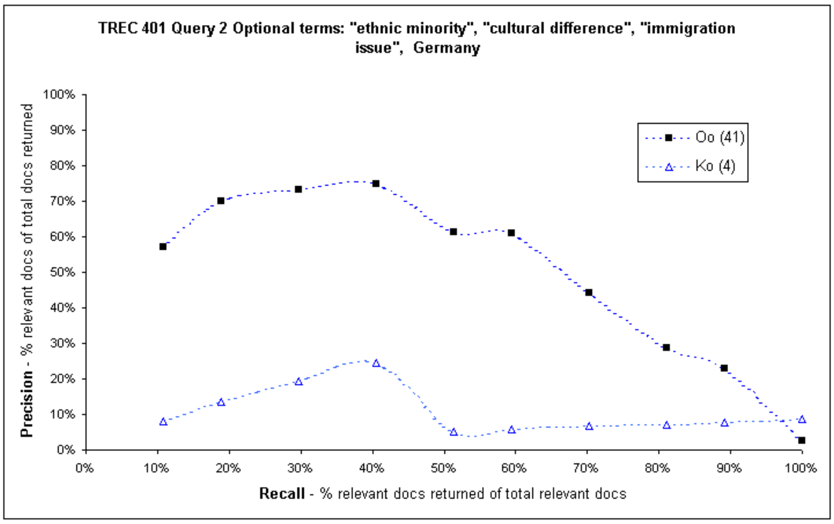


Fig. A18. T401 Q2 *optional* mode P&R.

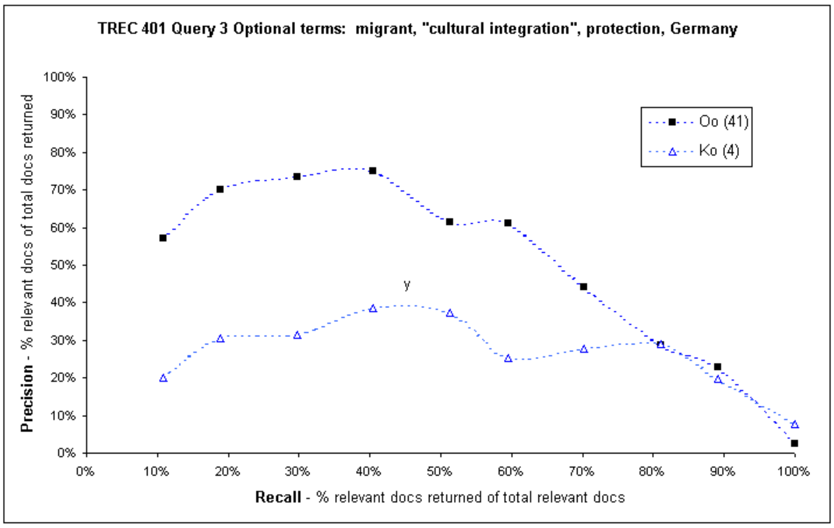


Fig. A19. T401 Q3 *optional* mode P&R.

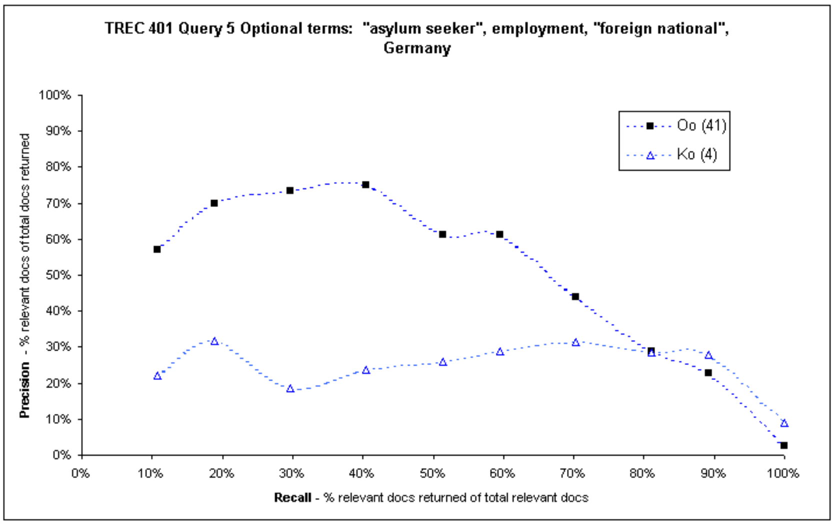


Fig. A20. T401 Q5 *optional* mode P&R.

T401 *Must-have* Mode P&R Graphs (not shown in main body).

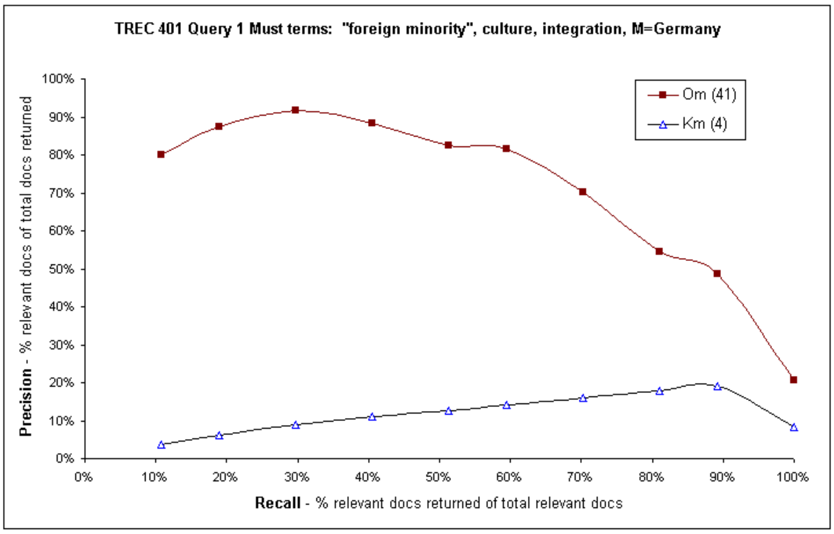


Fig. A21. T401 Q1 *must-have* mode P&R.

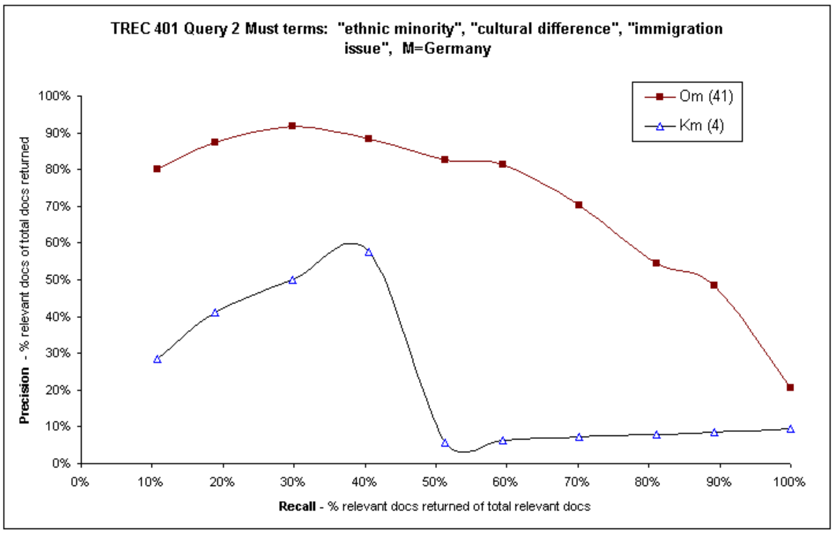


Fig. A22. T401 Q2 *must-have* mode P&R.

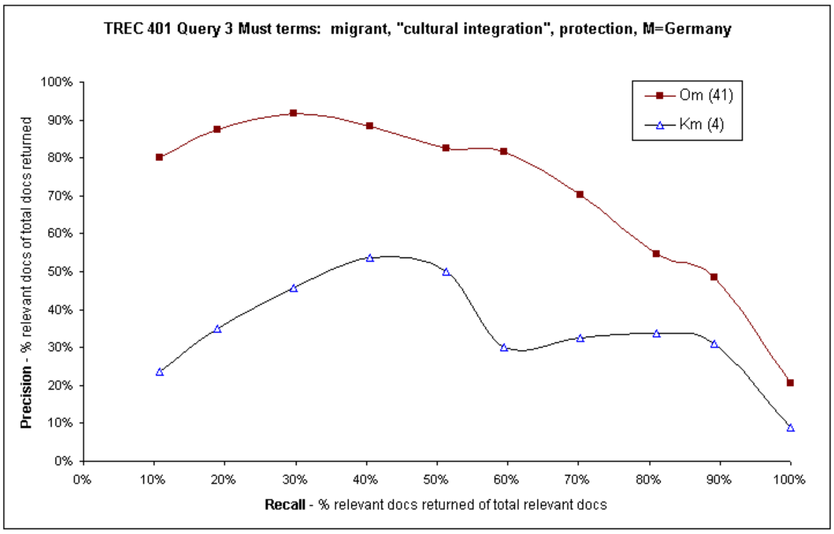


Fig. A23. T401 Q3 *must-have* mode P&R.

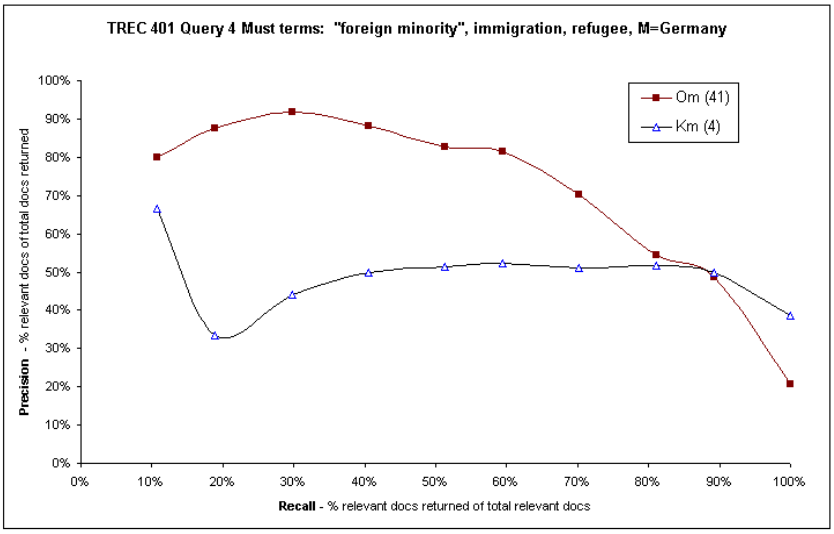


Fig. A24. T401 Q4 *must-have* mode P&R.

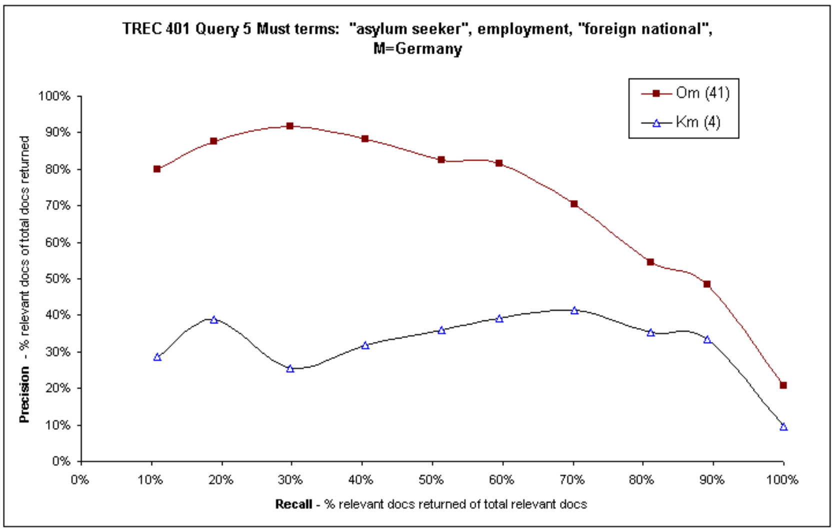


Fig. A25. T401 Q5 *must-have* mode P&R.

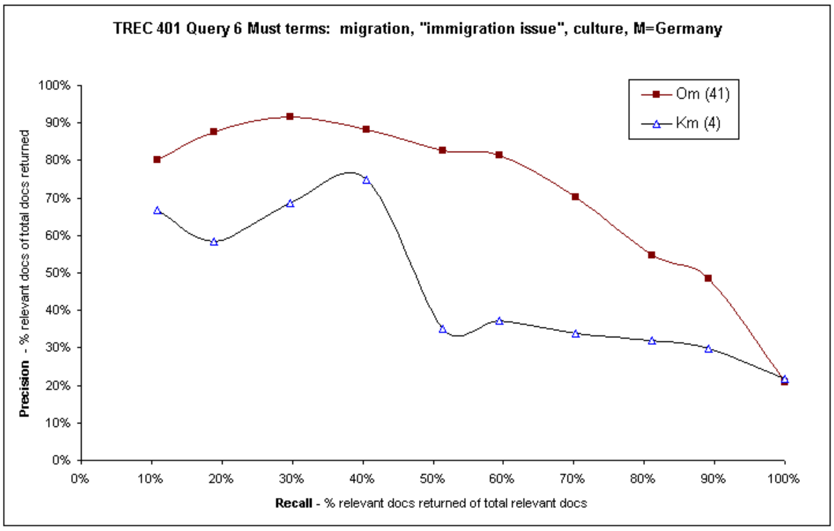


Fig. A26. T401 Q6 *must-have* mode P&R.

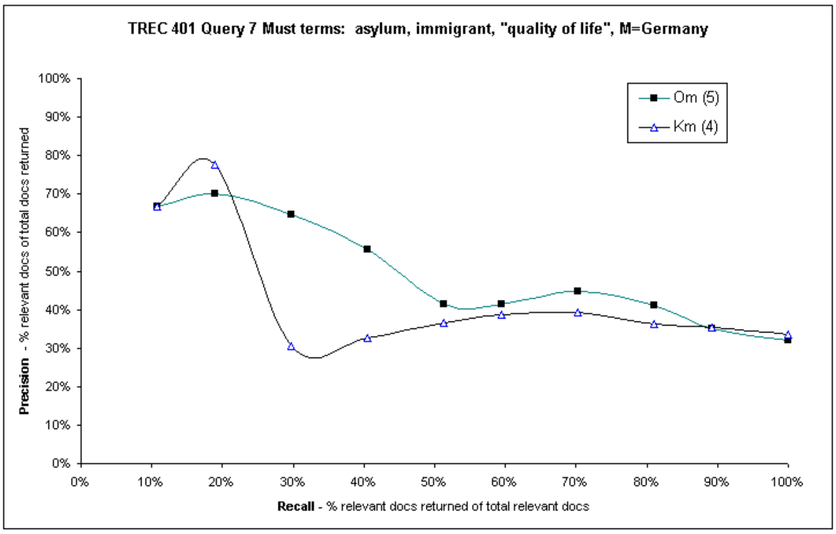


Fig. A27. T401 Q7 *must-have* mode P&R.

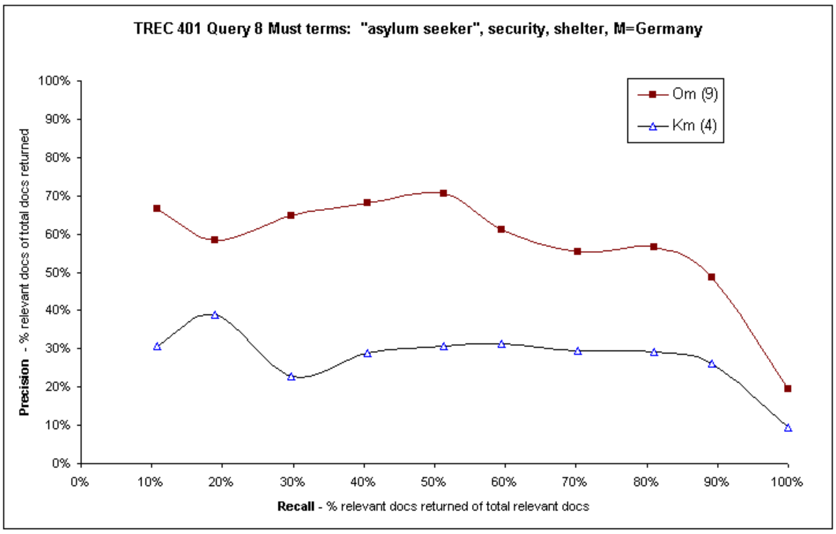


Fig. A28. T401 Q8 *must-have* mode P&R.

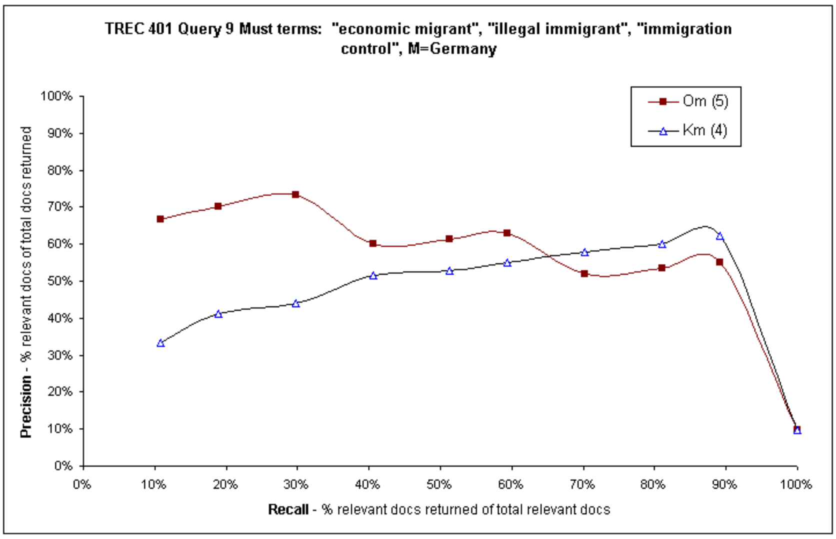


Fig. A29. T401 Q9 *must-have* mode P&R.

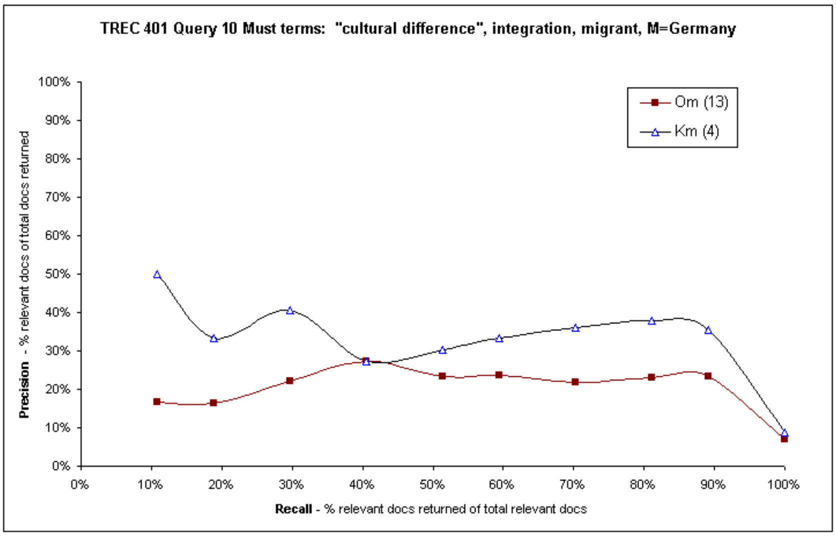


Fig. A30. T401 Q10 *must-have* mode P&R.

T416 *Optional* Mode P&R Graphs (not shown in main body).

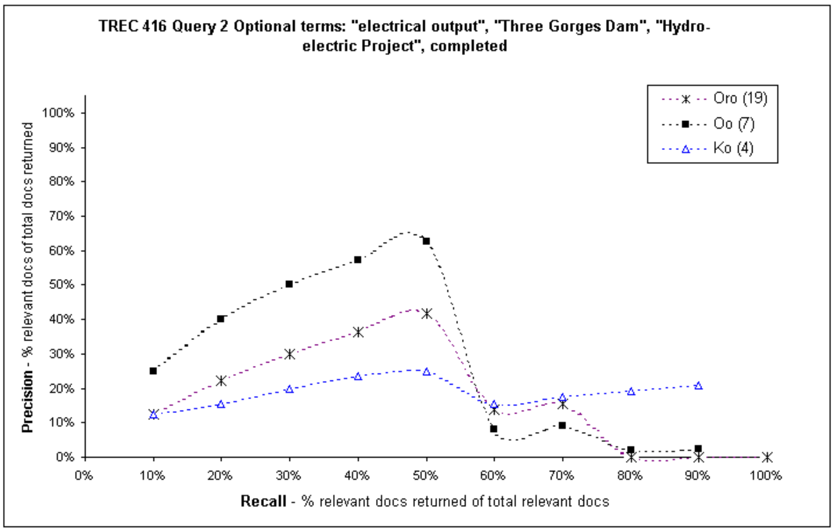


Fig. A31. T416 Q2 *optional* mode P&R.

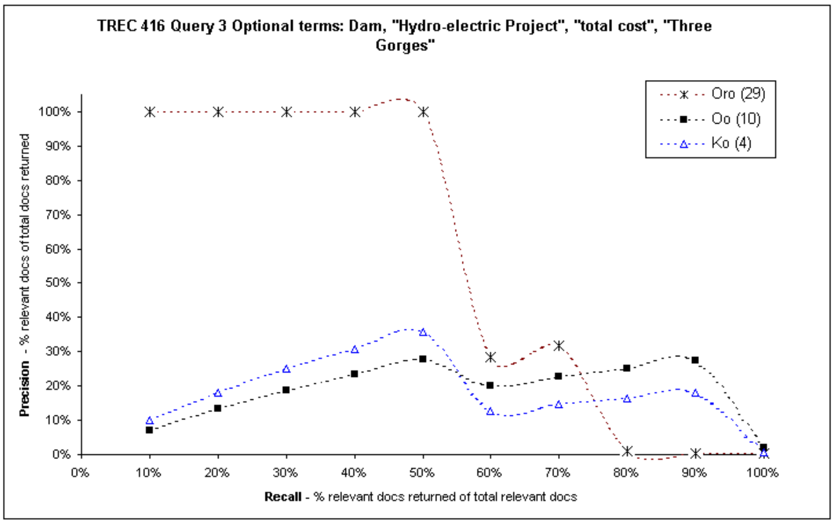


Fig. A32. T416 Q3 *optional* mode P&R.

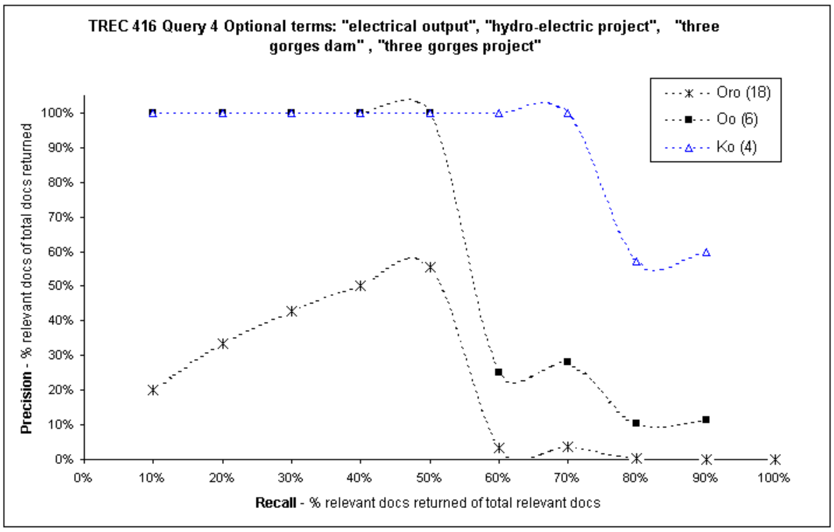


Fig. A33. T416 Q4 *optional* mode P&R.

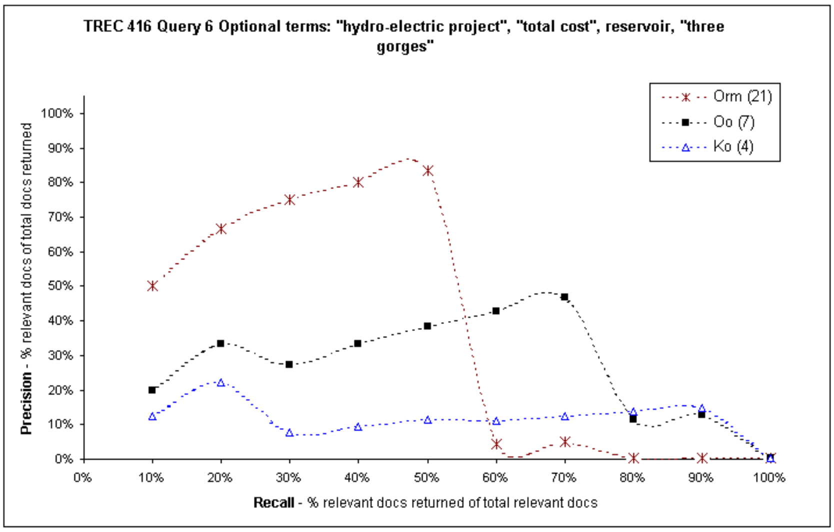


Fig. A34. T416 Q6 *optional* mode P&R.

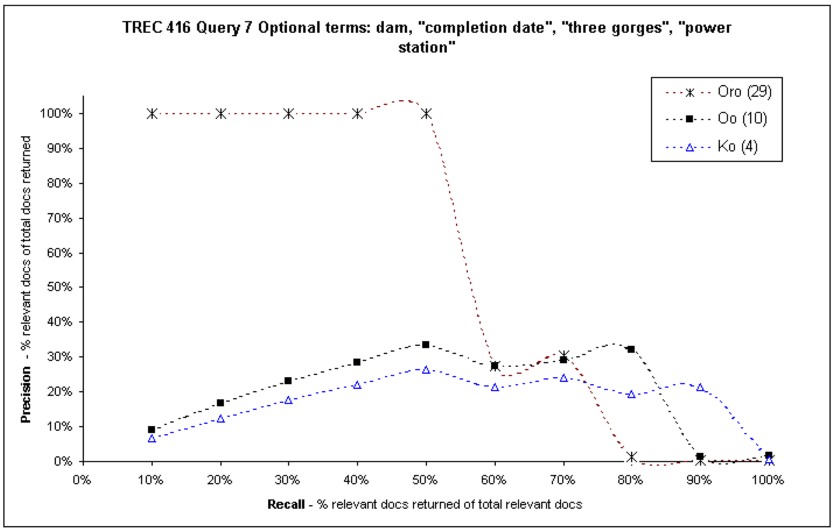


Fig. A35. T416 Q7 *optional* mode P&R.

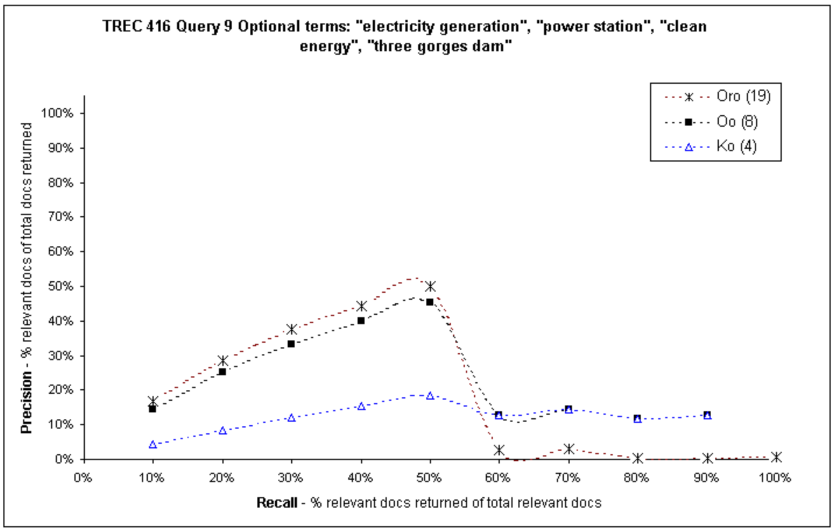


Fig. A36. T416 Q9 *optional* mode P&R.

T416 *Must-have* Mode P&R Graphs (not shown in main body).

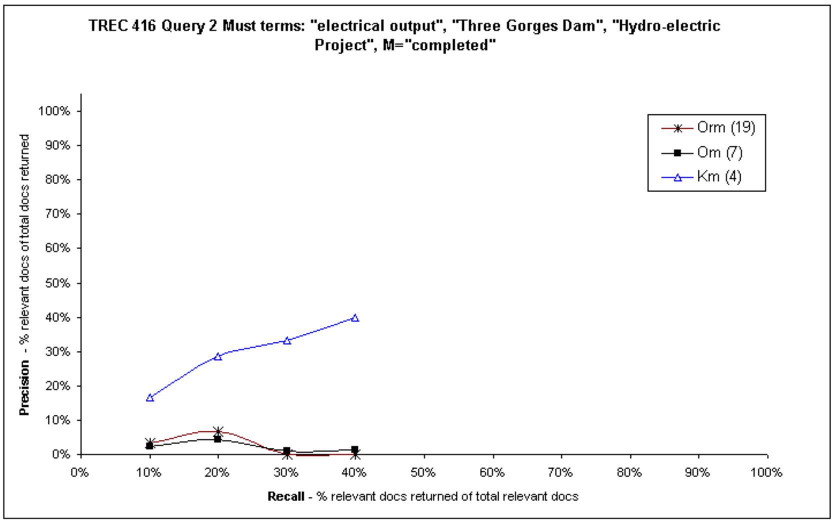


Fig. A37. T416 Q2 *must-have* mode P&R.

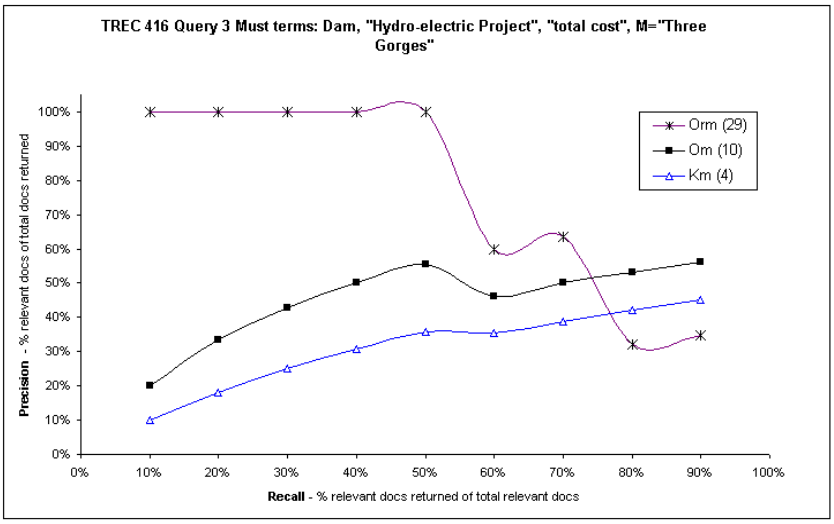


Fig. A38. T416 Q3 *must-have* mode P&R.

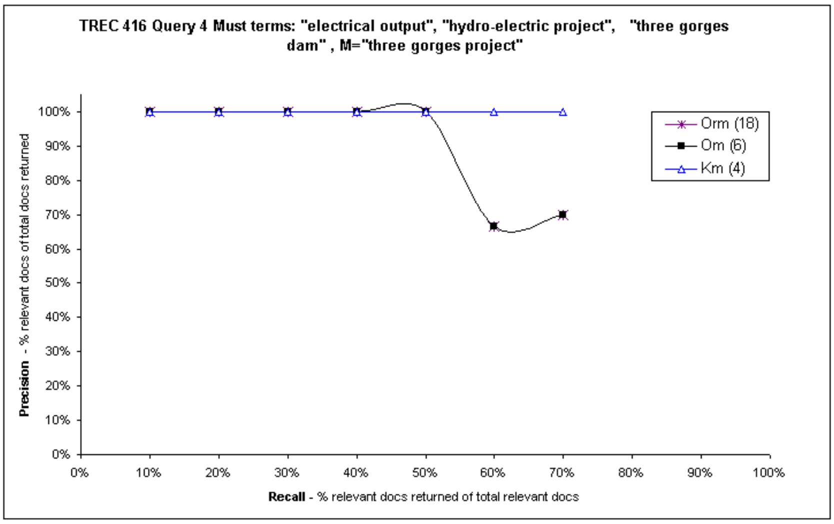


Fig. A39. T416 Q4 *must-have* mode P&R.

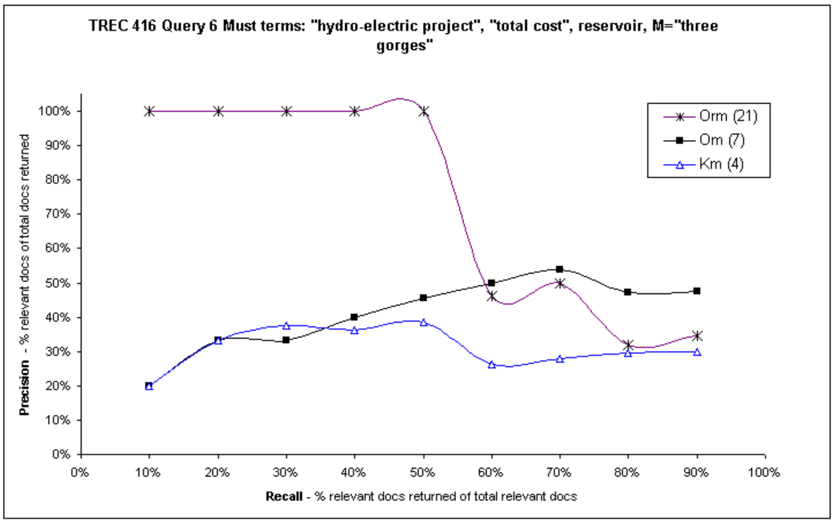


Fig. A40. T416 Q6 *must-have* mode P&R.

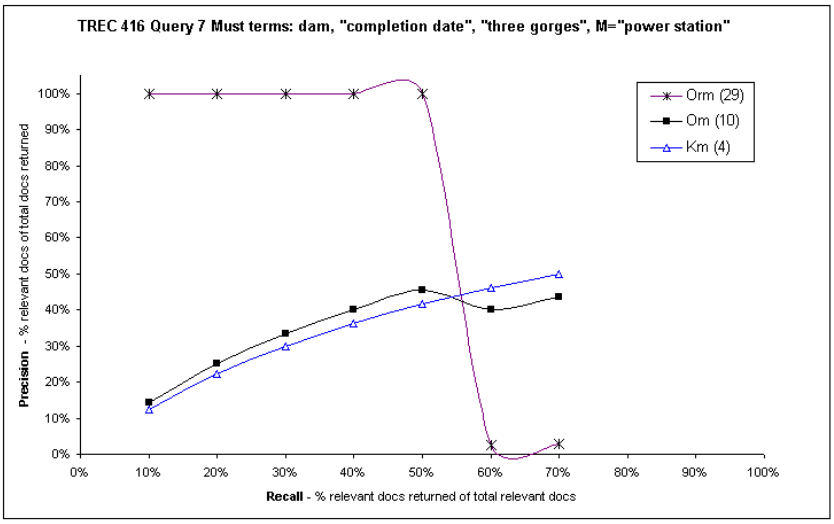


Fig. A41. T416 Q7 *must-have* mode P&R.

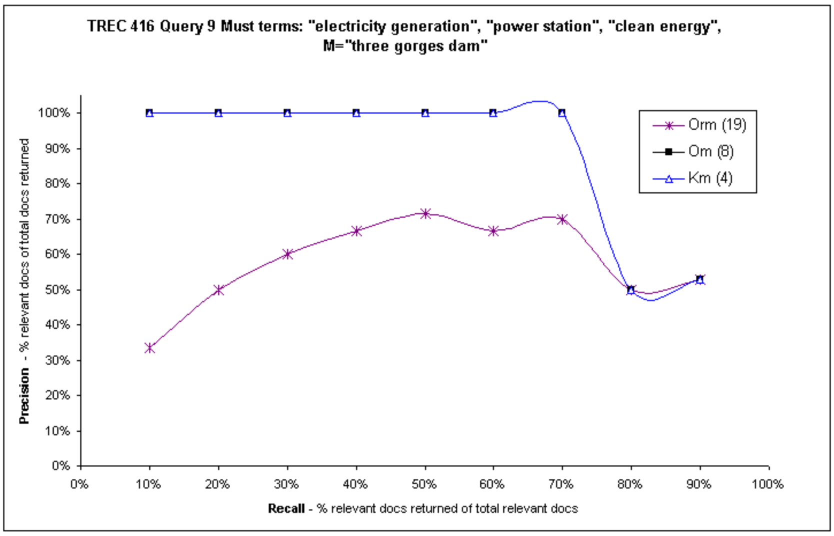


Fig. A42. T416 Q9 *must-have* mode P&R.

T438 *Optional* Mode P&R Graphs (not shown in main body).

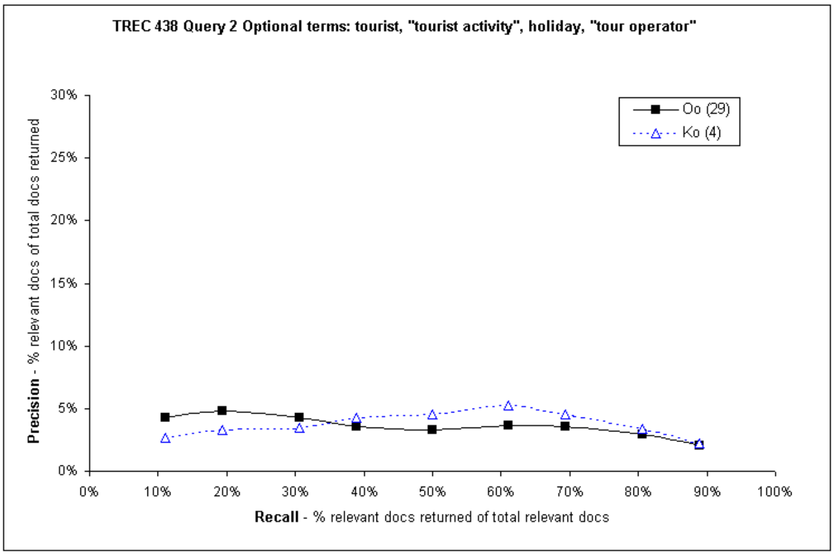


Fig. A43. T438 Q2 *optional* mode P&R.

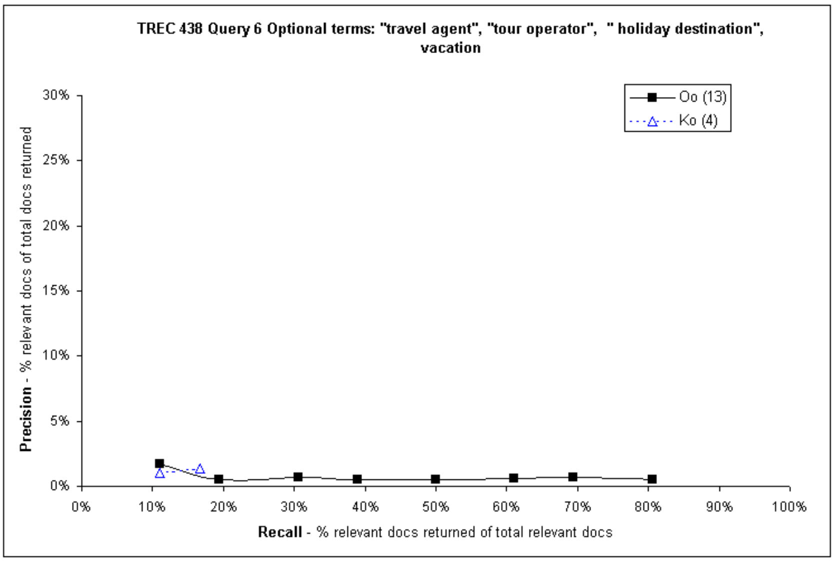


Fig. A44. T438 Q6 *optional* mode P&R.

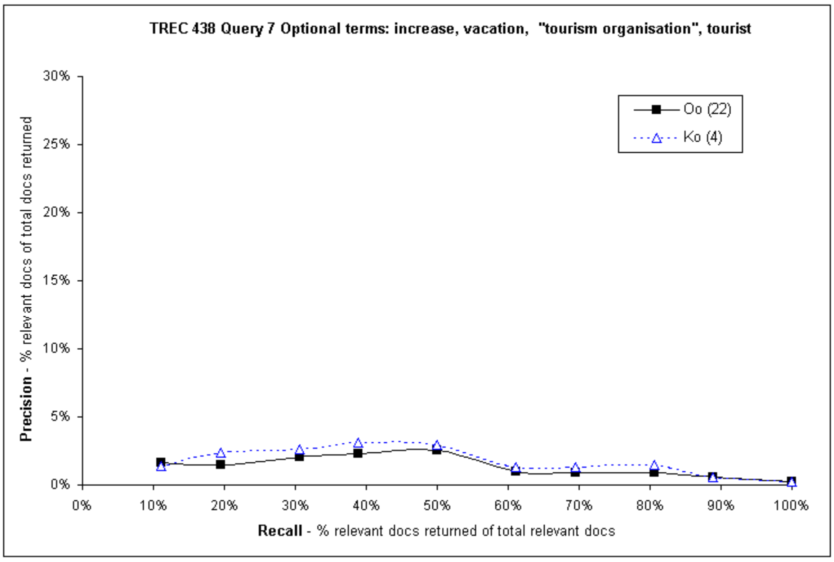


Fig. A45. T438 Q7 *optional* mode P&R.

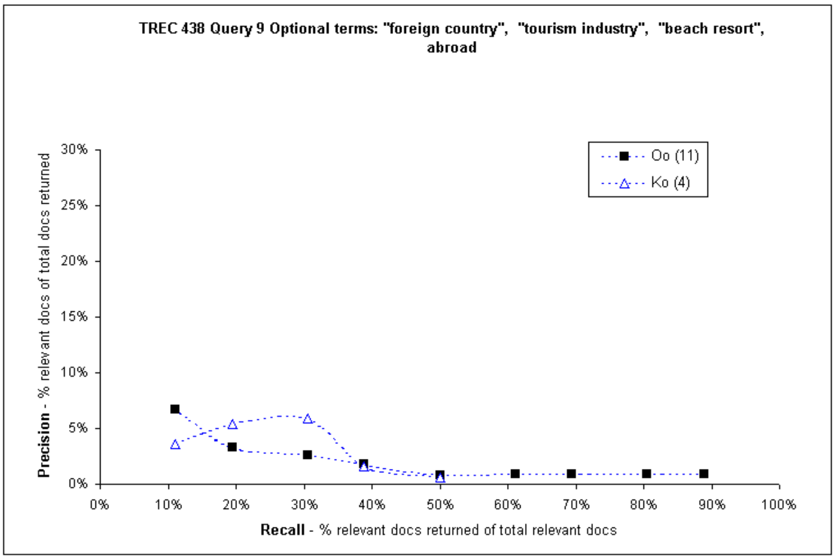


Fig. A46. T438 Q9 *optional* mode P&R.

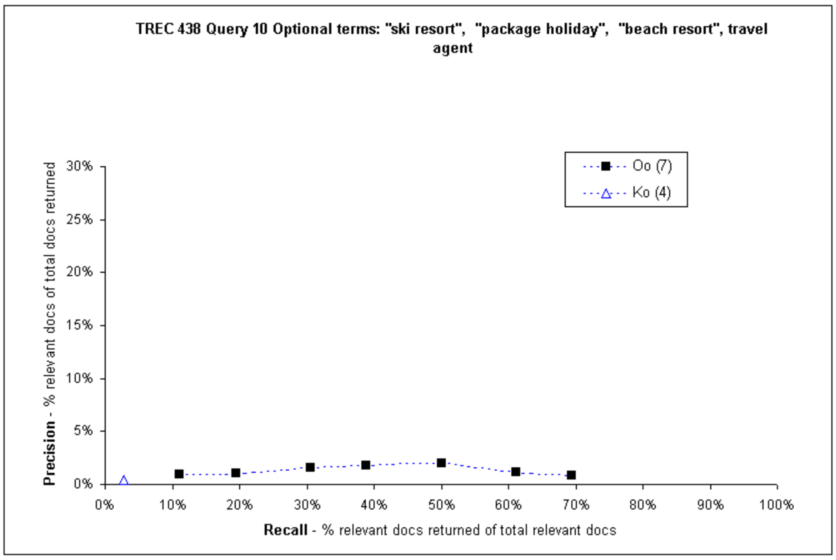


Fig. A47. T438 Q10 *optional* mode P&R.

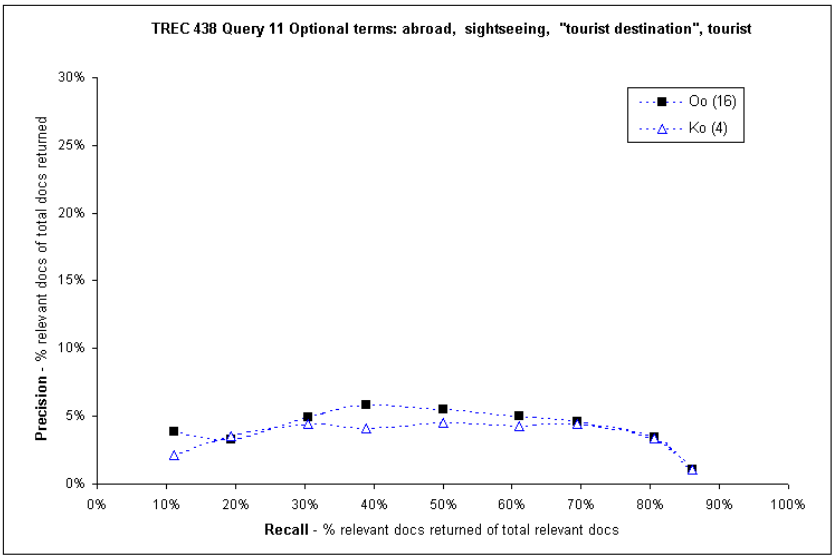


Fig. A48. T438 Q11 *optional* mode P&R.

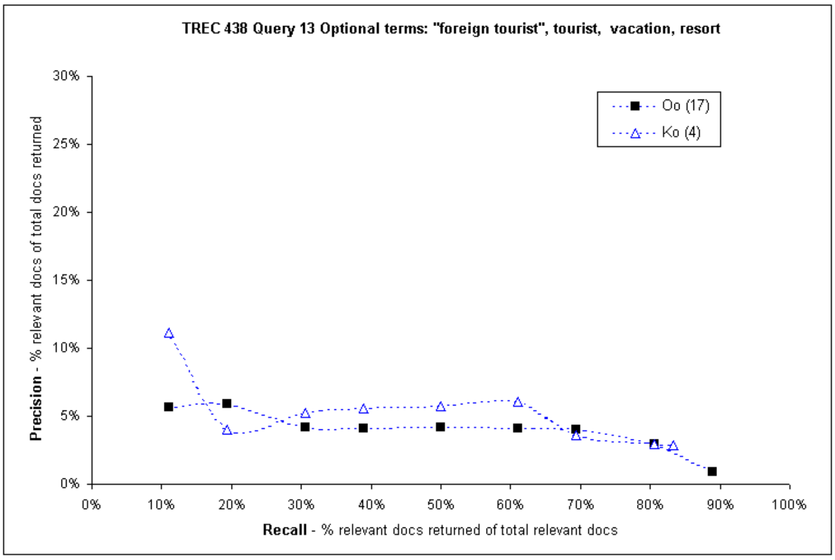


Fig. A49. T438 Q13 *optional* mode P&R.

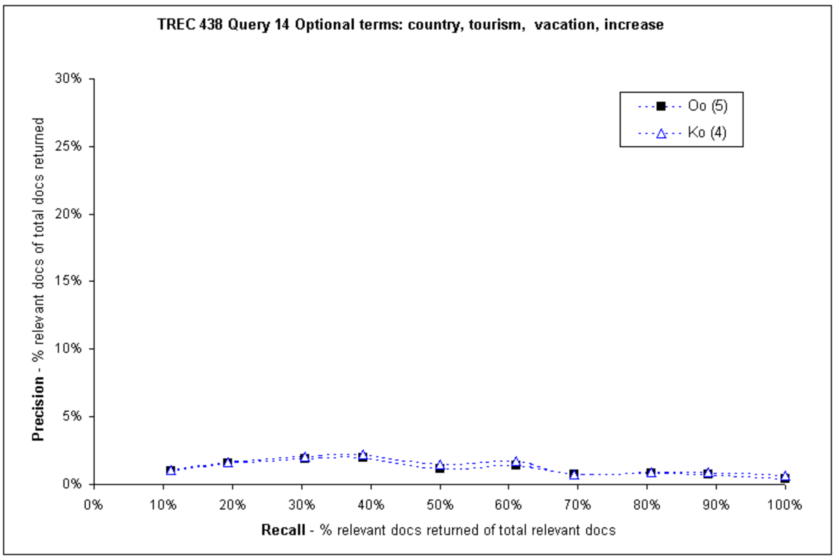


Fig. A50. T438 Q14 *optional* mode P&R.

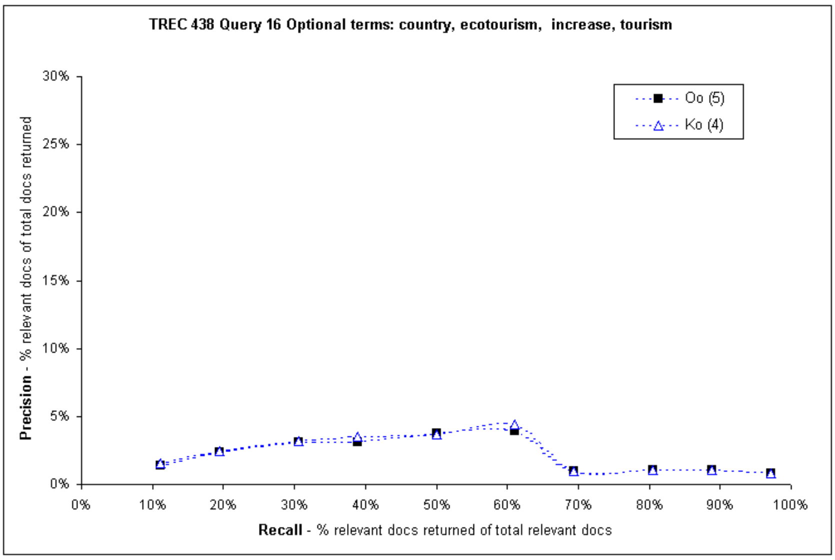


Fig. A51. T438 Q16 *optional* mode P&R.

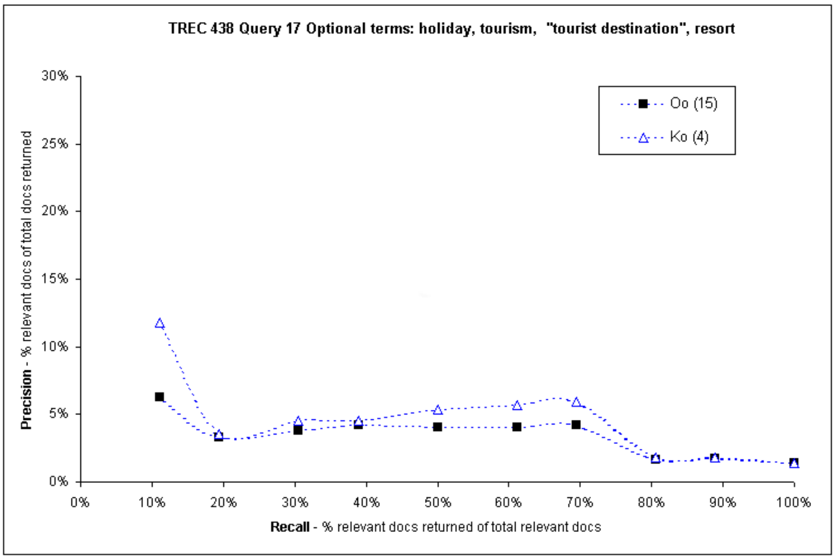


Fig. A52. T438 Q17 *optional* mode P&R.

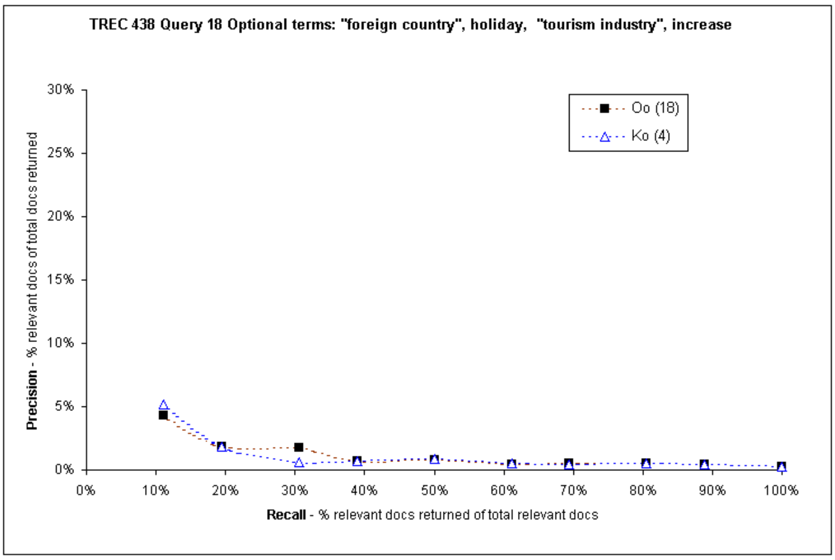


Fig. A53. T438 Q18 *optional* mode P&R.

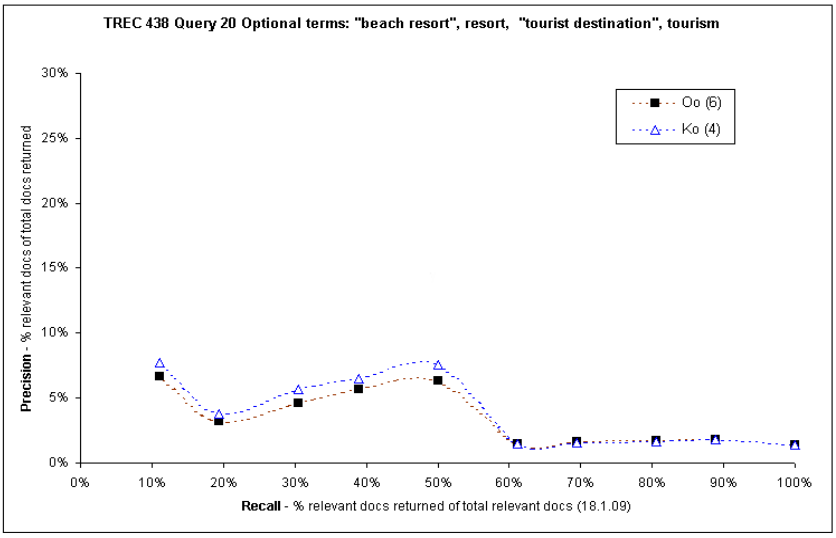


Fig. A54. T438 Q20 *optional* mode P&R.

T438 *Must-have* Mode P&R Graphs (not shown in main body).

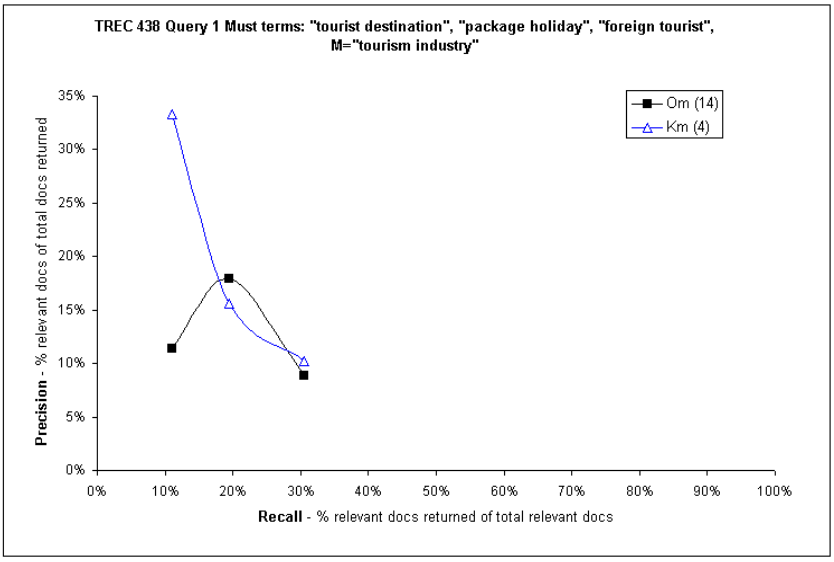


Fig. A55. T438 Q1 *must-have* mode P&R.

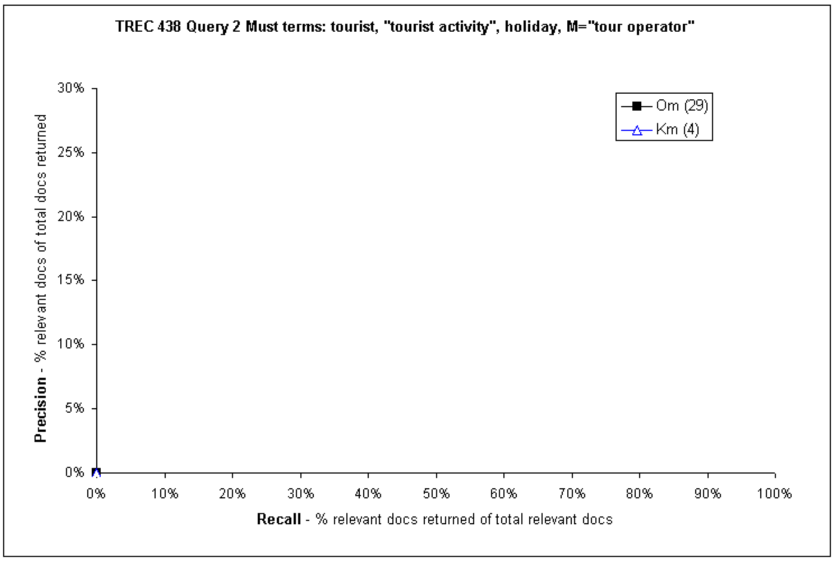


Fig. A56. T438 Q2 *must-have* mode P&R.

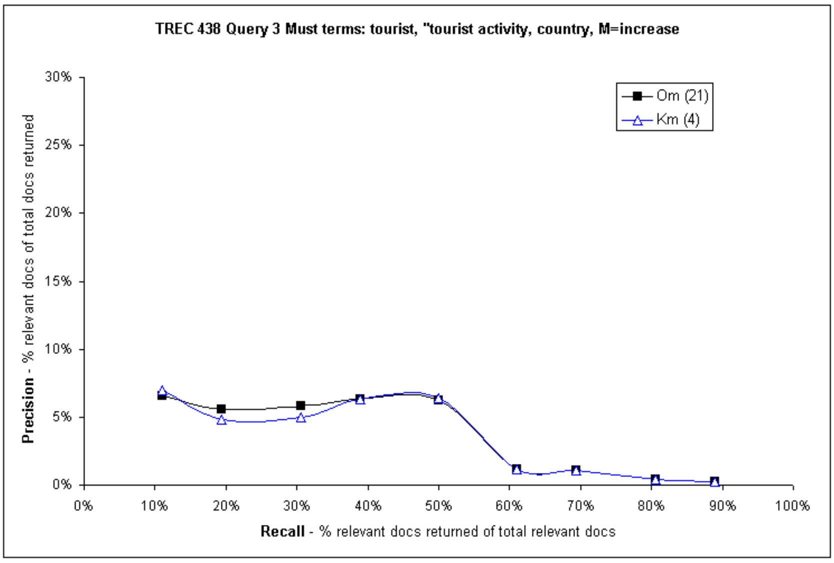


Fig. A57. T438 Q3 *must-have* mode P&R.

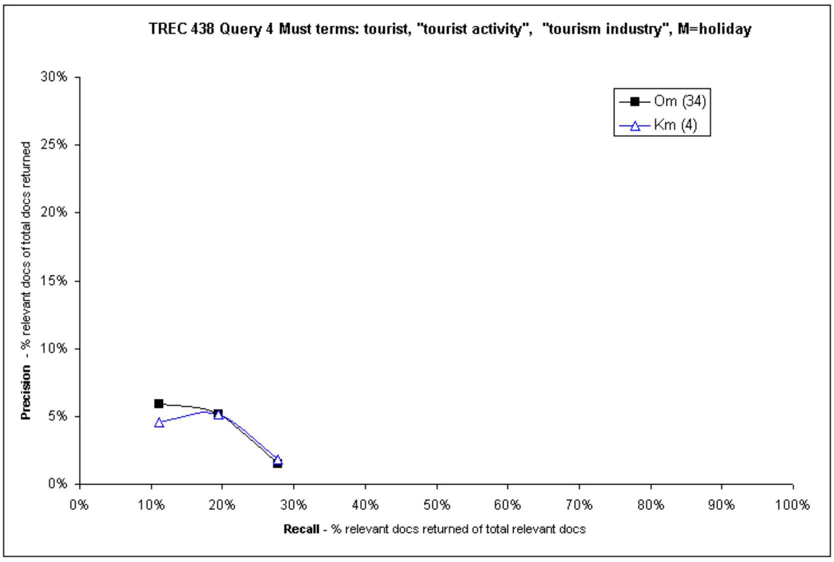


Fig. A58. T438 Q4 *must-have* mode P&R.

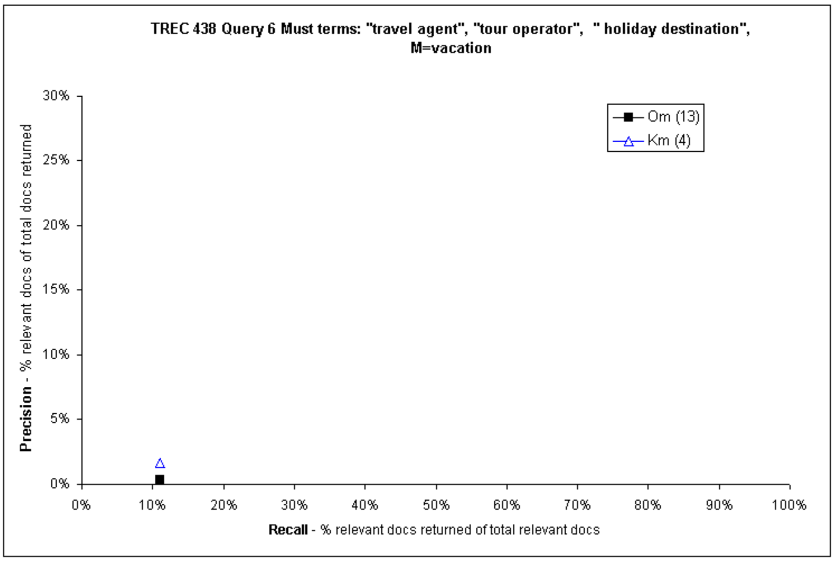


Fig. A59. T438 Q6 *must-have* mode P&R.

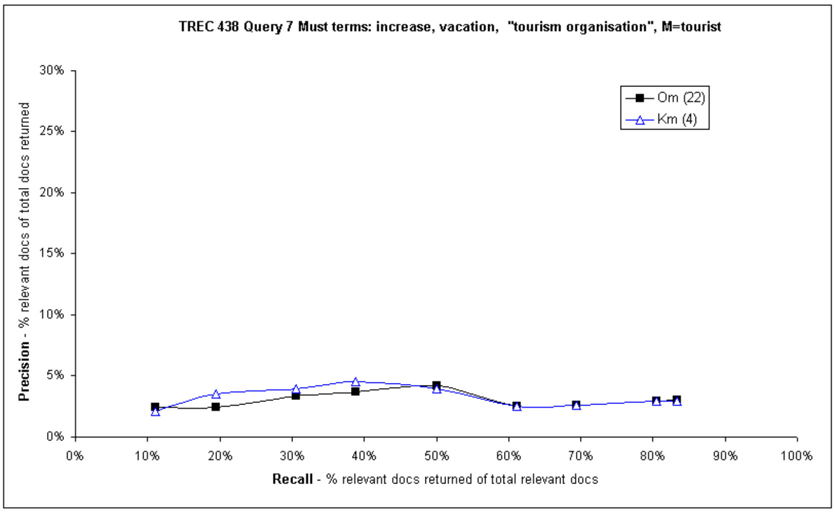


Fig. A60. T438 Q7 *must-have* mode P&R.

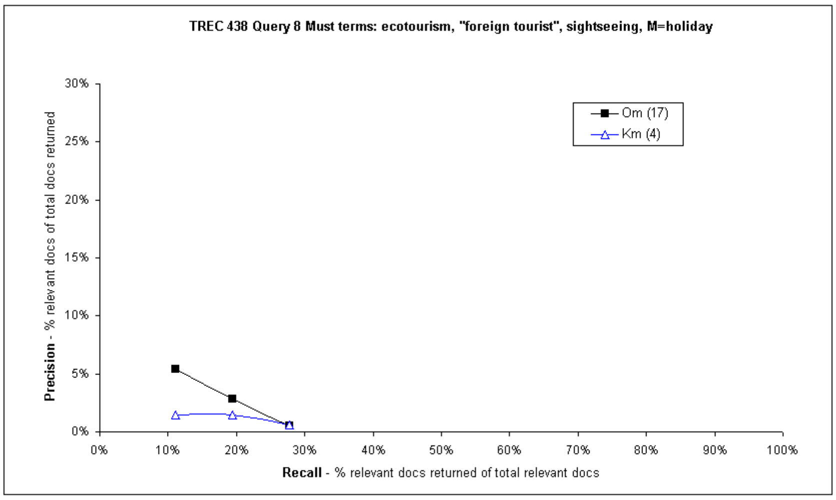


Fig. A61. T438 Q8 *must-have* mode P&R.

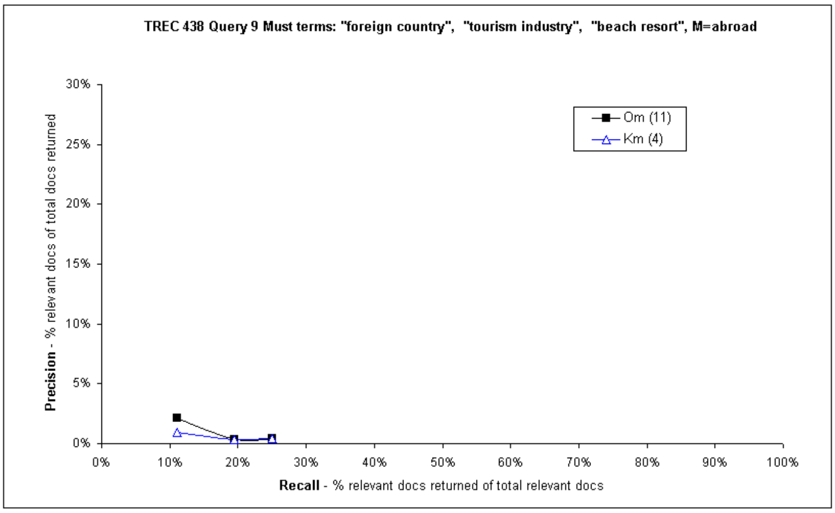


Fig. A62. T438 Q9 *must-have* mode P&R.

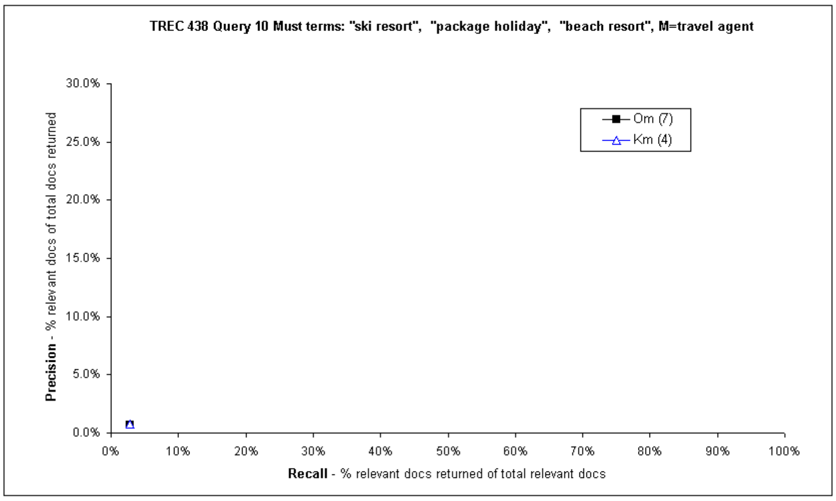


Fig. A63. T438 Q10 *must-have* mode P&R.

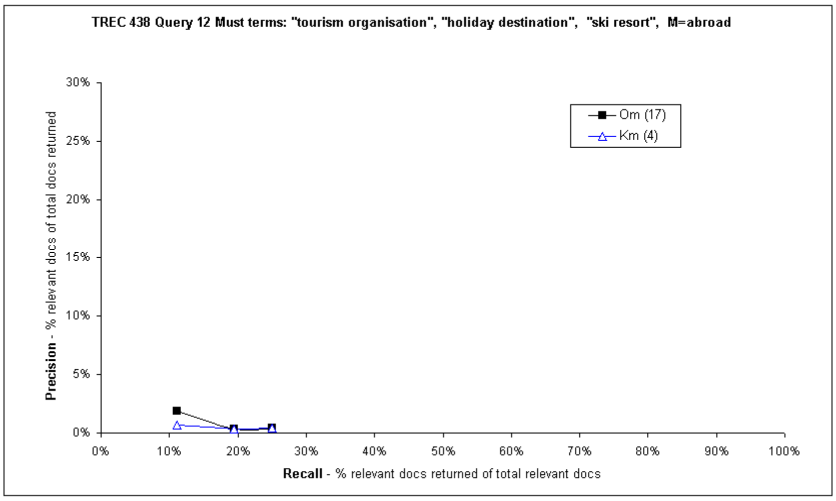


Fig. A64. T438 Q12 *must-have* mode P&R.

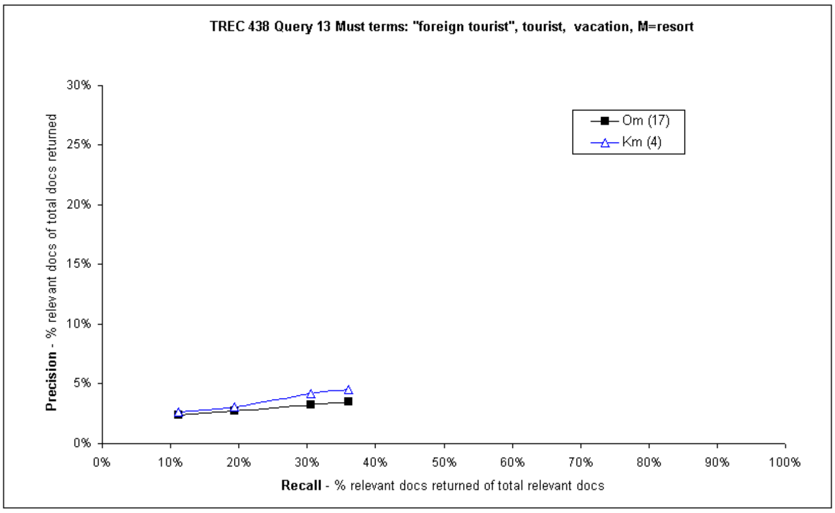


Fig. A65. T438 Q13 *must-have* mode P&R.

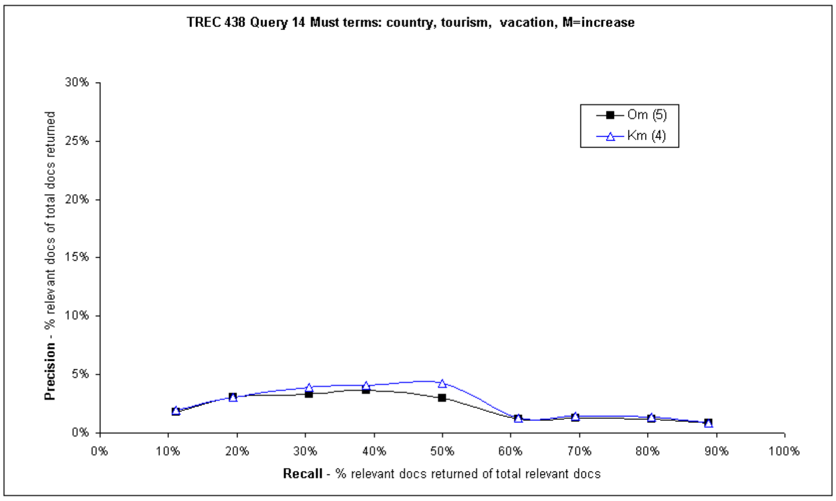


Fig. A66. T438 Q14 *must-have* mode P&R.

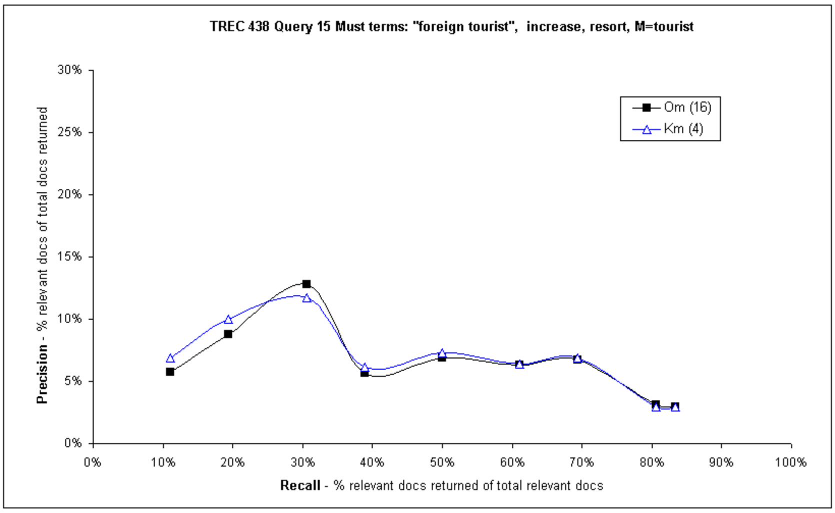


Fig. A67. T438 Q15 *must-have* mode P&R.

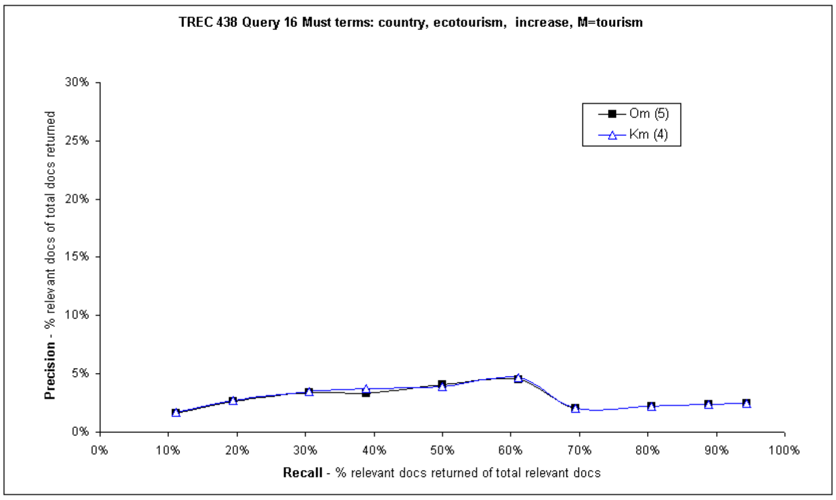


Fig. A68. T438 Q16 *must-have* mode P&R.

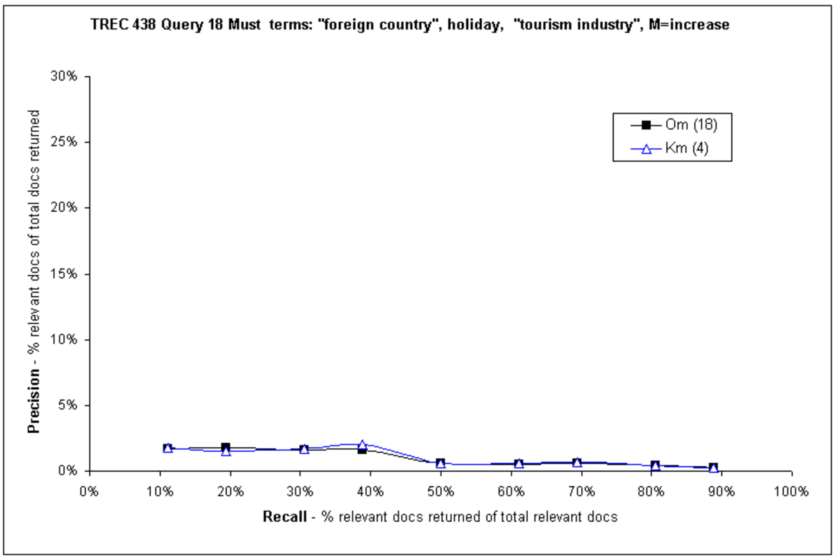


Fig. A69. T438 Q18 *must-have* mode P&R.

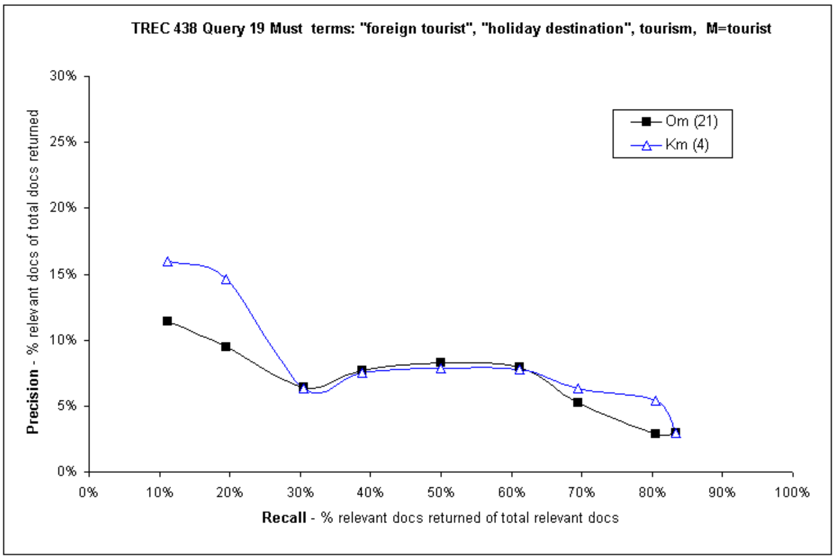


Fig. A70. T438 Q19 *must-have* mode P&R.

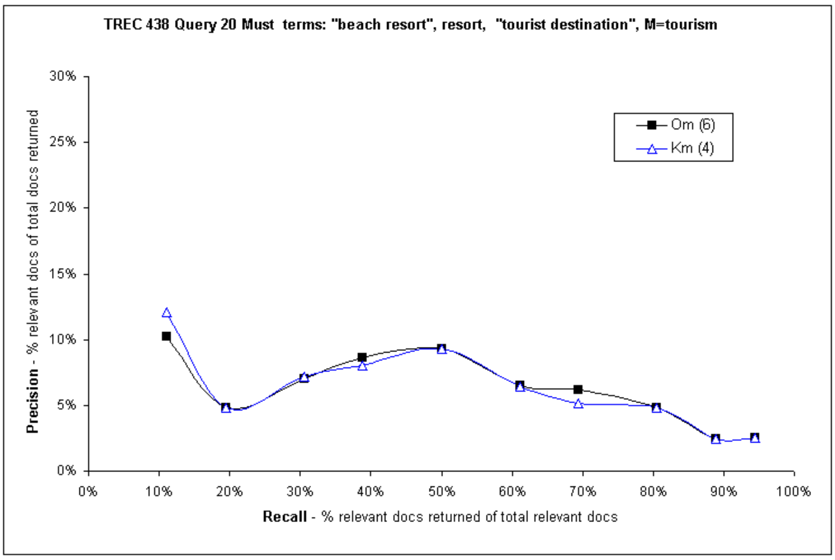


Fig. A71. T438 Q20 *must-have* mode P&R.

APPENDIX H: EXAMPLE OF RETRIEVED QUERY DATA

Data: T416 Hydro-electric query 4 (keyword-only on full TREC corpus - 14 relevant docs).

```
VSM [tf-idf] Sort:      2009.08.11 09:59:28
hydro-electric_T416_ <Q4_Keywords_Max>

VSM Rankings
DocNumber          tf-idf score       Relv Doc    Tot Docs

WT03-B36-2         19.0211              1           1
WT02-B36-12        19.0211              2           2
WT19-B17-157       14.0540              3           3
WT16-B31-94        14.0540              4           4
WT20-B21-16        14.0540              5           5
WT16-B06-1         14.0540              6           6
WT15-B12-136       14.0540              7           7
WT07-B30-100       14.0540              8           8
WT26-B32-117       14.0540              9           9
WT20-B01-141       11.0328
WT19-B39-31        11.0328
WT01-B09-202       11.0328
WT17-B14-31        11.0328
WT08-B34-62        9.9342
WT08-B34-61        9.9342
WT08-B34-42        9.9342
WT08-B26-57        9.0869
WT08-B18-175       9.0869
WT08-B08-147       9.0869
WT09-B16-120       9.0869
WT10-B23-15        9.0869
WT10-B30-161       9.0869
WT11-B30-83        9.0869
WT12-B11-137       9.0869
WT12-B11-142       9.0869
WT12-B12-4         9.0869
WT12-B37-263       9.0869
WT13-B05-11        9.0869
WT14-B07-61        9.0869
WT14-B21-7         9.0869
WT14-B35-11        9.0869
WT14-B35-14        9.0869
WT14-B35-15        9.0869
WT15-B07-163       9.0869
WT04-B16-520       9.0869
WT15-B22-189       9.0869
WT04-B03-1         9.0869               10          37
WT16-B15-181       9.0869
WT03-B37-6         9.0869
WT17-B01-68        9.0869
WT17-B13-26        9.0869
WT03-B20-85        9.0869
WT18-B30-80        9.0869
WT03-B03-10        9.0869               11          44
WT19-B37-177       9.0869
WT02-B37-20        9.0869
WT02-B16-260       9.0869
WT02-B16-247       9.0869
WT20-B30-106       9.0869
WT21-B04-160       9.0869
WT21-B25-43        9.0869
WT21-B38-66        9.0869
WT22-B31-59        9.0869               12          53
WT22-B36-106       9.0869
WT23-B39-355       9.0869
WT23-B40-26        9.0869
WT23-B40-30        9.0869
WT23-B40-33        9.0869
WT25-B20-88        9.0869               13          59
WT01-B03-75        9.0869
WT27-B02-17        9.0869
WT27-B21-503       9.0869
WT27-B21-507       9.0869
```

```
VSM Docs:          2009.08.11 09:59:28 [63]

[No] Query terms found [No. Docs]:

[1] electrical output  [28]
[2] hydro-electric project  [4]
[3] three gorges dam  [28]
[4] three gorges project  [12]


P&R Summary                      10% Rec Pts    Tot Docs

WT03-B36-2         19.0211           1            1
WT19-B17-157       14.0540           3            3
WT16-B31-94        14.0540           4            4
WT16-B06-1         14.0540           6            6
WT15-B12-136       14.0540           7            7
WT07-B30-100       14.0540           8            8
WT04-B03-1         9.0869            10           37
WT03-B03-10        9.0869            11           44
WT25-B20-88        9.0869            13           59
```

Note: the P&R summary presents the cumulative number of relevant documents recalled, and the cumulative number of all documents returned, at each 10% recall point, based on the number of relevant documents (maximum of 14) in the corpus.

E.g. at 70% recall: 10 relevant documents were recalled from 37 returned.

APPENDIX I: AVERAGE PERCENTAGE PRECISION VALUES (APV)

Average Precision Value (APV) calculated across 10% to 30% Recall Points - Fig. nos. denote P&R Graphs discussed in Results section 4

T401 - section 4.1

		K	OQE	
All OQE	Q1 Opt	5%	67%	
	Q1 Must	6%	86%	
	Q2 Opt	14%	67%	
	Q2 Must	40%	86%	
	Q3 Opt	27%	67%	
	Q3 Must	35%	86%	
	Q4 Opt	44%	67%	Fig. 72
	Q4 Must	48%	86%	
	Q5Opt	24%	67%	
	Q5 Must	31%	86%	
	Q6Opt	65%	67%	Fig. 73
	Q6 Must	65%	86%	
S+S OQE	Q7 Opt	53%	61%	Fig. 75
	Q7 Must	58%	67%	
	Q8 Opt	23%	63%	Fig. 76
	Q8 Must	31%	63%	
	Q9 Opt	34%	59%	Fig. 77
	Q9 Must	40%	70%	
	Q10 Opt	33%	13%	Fig. 78
	Q10 Must	41%	18%	

All Opt	18	46	Fig. 84	Fig. 128
All Opt MEA	32	60	Fig. 85	
All Must	25	59	Fig. 86	
All Must MEA	39	74	Fig. 87	
Q1-6 Opt	14	67	Fig. 70	
Q1-6 Opt MEA	30	67	Fig. 71	
Q1-6 Must	21	86	Fig. 80	
Q1-6 Must MEA	37	86	Fig. 81	
Q7-10 Opt	32	31	Fig. 74	
Q7-10 Opt MEA	35	49	Fig. 79	
Q7-10 Must	39	40	Fig. 82	
Q7-10 Must MEA	42	55	Fig. 83	

T416 - section 4.2

	K	S+S OQE	S+S+R OQE	
Q1 Opt	15%	28%	100%	Fig. 95
Q1 Must	24%	48%	100%	Fig. 96
Q2 Opt	16%	38%	22%	
Q2 Must	26%	2%	3%	
Q3 Opt	18%	13%	100%	
Q3 Must	18%	32%	100%	
Q4 Opt	100%	100%	32%	
Q4 Must	100%	100%	100%	
Q5 Opt	64%	64%	24%	Fig. 99
Q5 Must	100%	100%	48%	Fig. 100
Q6 Opt	14%	27%	64%	
Q6 Must	30%	29%	100%	
Q7 Opt	12%	16%	100%	
Q7 Must	22%	24%	100%	
Q8 Opt	19%	38%	64%	Fig. 97
Q8 Must	22%	38%	64%	Fig. 98
Q9 Opt	8%	24%	28%	
Q9 Must	100%	100%	48%	
Q10 Opt	7%	7%	7%	Fig. 101
Q10 Must	48%	48%	48%	Fig. 102

All Opt	14	21	27	Fig. 90	Fig. 129
All Opt MEA	27	35	54	Fig. 91	
All Must	32	15	20	Fig. 92	
All Must-Rev	32	22	34	Fig. 93	
All Must MEA	49	52	71	Fig. 94	

T438 - section 4.3

	K	S+S OQE	
Q1 Opt	21.2%	12.2%	Fig. 117
Q1 Must	19.7%	12.7%	
Q2 Opt	3.1%	4.5%	
Q2 Must	0.0%	0.0%	
Q3 Opt	4.4%	4.1%	Fig. 113
Q3 Must	5.6%	6.0%	
Q4 Opt	7.6%	9.2%	Fig. 114
Q4 Must	3.8%	4.2%	
Q5Opt	0.7%	1.8%	Fig. 109
Q5 Must	0.6%	1.1%	Fig. 116
Q6Opt	1.2%	1.0%	
Q6 Must	1.6%	0.4%	
Q7 Opt	2.1%	1.7%	
Q7 Must	3.2%	2.7%	
Q8 Opt	9.9%	6.2%	Fig. 118
Q8 Must	1.2%	2.9%	
Q9 Opt	5.0%	4.2%	
Q9 Must	0.6%	1.0%	
Q10 Opt	0.4%	1.2%	
Q10 Must	0.8%	0.7%	
Q11 Opt	3.4%	4.0%	
Q11 Must	4.3%	4.8%	Fig. 115
Q12 Opt	0.5%	2.7%	Fig. 110
Q12 Must	0.5%	0.9%	
Q13 Opt	6.8%	5.2%	
Q13 Must	3.3%	2.8%	
Q14 Opt	1.6%	1.5%	
Q14 Must	3.0%	2.7%	
Q15Opt	9.5%	8.3%	Fig. 111
Q15 Must	9.5%	9.1%	
Q16Opt	2.4%	2.3%	
Q16 Must	2.6%	2.5%	
Q17 Opt	6.6%	4.5%	
Q17 Must	6.7%	4.2%	Fig. 119
Q18 Opt	2.6%	2.6%	
Q18 Must	1.7%	1.7%	
Q19 Opt	12.0%	9.1%	Fig. 112
Q19 Must	12.3%	9.1%	
Q20 Opt	5.7%	4.8%	
Q20 Must	8.0%	7.4%	

	K	S+S OQE		
All Opt	2.1	2.6	Fig. 105	Fig. 130
All Opt MEA	5.6	4.7	Fig. 106	
All Must	1.7	1.7	Fig. 107	
All Must MEA	4.9	4.3	Fig. 108	

T401 Weight Variations - section 4.4

T401 K v Wtd Oros	K	1,0.5,0.3,0.2	1,0.7,0.5,0.3	1,0.9,0.7,0.5	1,1,1,1	
Q10 Opt	33%	16%	30%	47%	47%	Fig. 122

T401 K v Wtd Oros	K	StdWt Oro	RevWt Oro	NonWt Oro	
Q7-10 Opt	32	53	56	59	Fig. 123

T401 K v Wtd Oos	K	StdWt Oro	RevWt Oro	ExtdWt Oro	
All Opt	18	15	18	20	Fig. 124

T401 versus T401+SUMO - section 4.4

T401 v T401+SUMO	T401	T401+SUMO	
All Opt (S+S)	14	12	Fig. 126
All Opt MEA (S+S)	40	38	Fig. 127

T438 S+S+R OQE - section 4.4

T438	K	S+S OQE	S+S+R OQE	
13Qs Opt	2.1	3.5	2.8	Fig. 131
13Qs Opt MEA	6.4	5.4	4.6	Fig. 132