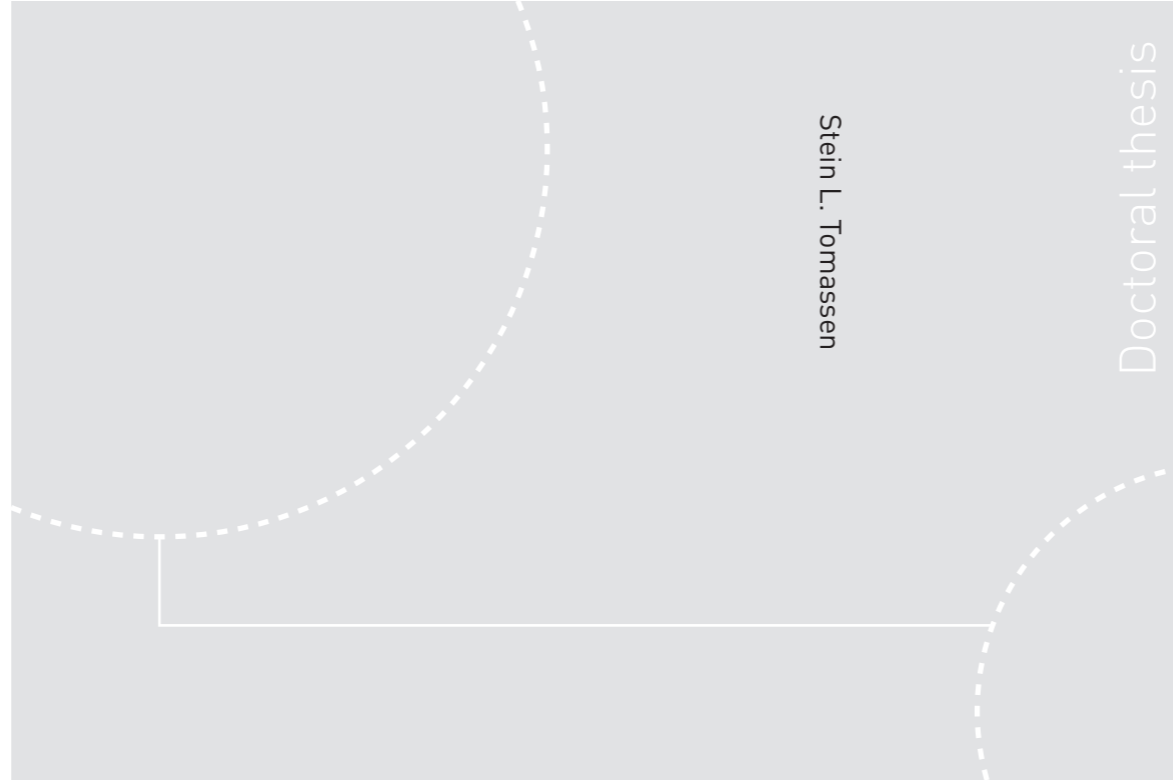


ISBN 978-82-471-2625-7 (printed ver.)
ISBN 978-82-471-2626-4 (electronic ver.)
ISSN 1503-8181



Doctoral theses at NTNU, 2011:51

Stein L. Tomassen

Conceptual Ontology Enrichment for Web Information Retrieval

Stein L. Tomassen

Conceptual Ontology Enrichment for Web Information Retrieval

Thesis for the degree of Philosophiae Doctor

Trondheim, Mars 2011

Norwegian University of Science and Technology
Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Computer and Information Science



NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology,
Mathematics and Electrical Engineering
Department of Computer and Information Science

© Stein L. Tomassen

ISBN 978-82-471-2625-7 (printed ver.)
ISBN 978-82-471-2626-4 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2011:51

Printed by NTNU-trykk

"You shall know a word by the company it keeps"

John Rupert Firth (English linguist, 1890-1960)

Abstract

Searching for information on the Web can be frustrating. One of the reasons is the ambiguity of words. The work presented in this thesis concentrates on how the effectiveness of standard information retrieval systems can be enhanced with semantic technologies like ontologies. Ontologies are knowledge models that can represent knowledge of any universe of discourse by describing how concepts of a domain are related. Creating and maintaining ontologies can be tedious and costly. However, we focus on reusing ontologies, rather than engineering, and on their applicability to improve the retrieval effectiveness of existing search systems.

The aim of this work is to find an effective approach for applying ontologies to existing search systems. The basic idea is that these ontologies can be used to tackle the problem of ambiguous words and hence improve the retrieval effectiveness. Our approach to semantic search builds on feature vectors (FV). The basic idea is to connect the (standardised) domain terminology encoded in an ontology to the actual terminology used in a text corpus. Therefore, we propose to associate every ontology entity (classes and individuals are called entities in this work) with a FV that is tailored to the actual terminology used in a text corpus like the Web. These FVs are created off-line and later used on-line to filter (i.e. to disambiguate search) and re-rank the search results from an underlying search system. This pragmatic approach is applicable to existing search systems since it only depends on extending the query and presentation components, in other words there is no need to alter either the indexing or the ranking components of the existing systems.

A set of experiments have been carried out and the results report on improvement by more than 10%. Furthermore, we have shown that the approach is neither dependent on highly specific queries nor on a collection comprised only of relevant documents. In addition, we have shown that the FVs are relatively persistent, i.e. little maintenance of the FVs is required.

In this work, we focus on the creation and evaluation of these feature vectors. As a result, a part of the contribution of this work is a framework for the construction of FVs. Furthermore, we have proposed a set of metrics to measure the quality of the created FVs. We have also provided a set of guidelines for optimal construction of feature vectors for different categories of ontologies.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfilment of the requirements for the degree of *Philosophiae Doctor (PhD)*.

This work has been conducted in the Department of Computer and Information Science (IDI) at the Faculty of Information Technology, Mathematics and Electrical Engineering (IME), NTNU, Trondheim.

Acknowledgements

The work presented in this thesis has been financed by The Research Council of Norway (NFR) as part of the project "Integrated Information Platform for Reservoir and Subsea Production Systems" (IIP). NFR project number 163457/S30.

First, I would like to thank my main supervisor Prof. Jon Atle Gulla and co-supervisors Dr. Robert Engels and Dr. Per Gunnar Auran for fruitful discussions, guidance, and all their help.

I would also like to thank my colleagues at IDI for their support and help and for providing a pleasant working environment. I would like to express special thanks to Dr. Darijus Strasunskas for a very fruitful collaboration that has led to many published papers and an international workshop. He has continuously supported my work and been a great source of help in formulating and structuring my thoughts. He has provided constructive criticism of my work and good guidance that allowed me to achieve a degree. I would also like to thank Jeanine Lilleng for fruitful discussions and for being co-author of a paper, and Dr. Sari Hakkarainen for her help in the early stages of my work. Also, I would like to thank Geir Solskinnsbakk for sharing his office with me, we have had many interesting discussions about various topics that has made my time particularly memorable.

I would also like to thank fellow members of the IS-group and students at NTNU for participating in some of my experiments. A special thanks to the administrative and technical staff at IDI for providing the necessary infrastructure, helping me with many practical issues, and for being helpful and friendly. I would also like to thank colleagues at Det Norske Veritas (DNV) and Computas for their discussions and help in the early phase of my work.

Finally, I am immensely grateful to my family for their enduring support throughout my PhD. I want to thank my mother Reidun for being so attentive and showing so much interest in my work all these years, even when it was not easy for you to be interested. A special thank to my wonderful wife Ida for her love, support, and understanding, my son Sander and my daughter Kajsa for their love, patience and joy that has provided me with the inspiration to fulfil this task. I would also like to express my gratitude to Malin who has brought much joy and happiness into our lives.

Stein Løkke Tomassen
08 March 2011

Contents

PART I RESEARCH CONTEXT AND RESULTS	1
1 INTRODUCTION	3
1.1 <i>Background and motivation</i>	3
1.2 <i>Problem outline</i>	7
1.3 <i>Research context</i>	9
1.4 <i>Objectives and research questions</i>	10
1.5 <i>Research approach and scope</i>	11
1.6 <i>Contributions</i>	12
1.7 <i>Overview of main publications</i>	14
1.8 <i>Thesis structure</i>	15
2 RESEARCH APPROACH	17
2.1 <i>Introduction</i>	17
2.2 <i>Empirical research methods</i>	17
2.3 <i>Overall research approach</i>	19
3 RELATED WORK	23
3.1 <i>Introduction</i>	23
3.2 <i>Semantics in Information Retrieval</i>	26
3.3 <i>Evaluation of semantic search systems</i>	44
3.4 <i>Summary</i>	48
4 RESULTS	51
4.1 <i>Feature Vectors</i>	51
4.2 <i>Implementations</i>	52
4.3 <i>Experiments</i>	57
4.4 <i>Synopsis of main publications</i>	67
5 EVALUATION	81
5.1 <i>Research questions revisited</i>	81
5.2 <i>Contributions</i>	83
5.3 <i>Contributions in relation to related work</i>	86
5.4 <i>Relevance of contributions</i>	87
5.5 <i>Validity discussion</i>	88
6 CONCLUSIONS AND FUTURE WORK	93
6.1 <i>Conclusions</i>	93
6.2 <i>Directions for future work</i>	94
REFERENCES.....	97
PART II PAPERS	103
P1: CONSTRUCTION OF ONTOLOGY BASED SEMANTIC-LINGUISTIC FEATURE VECTORS FOR SEARCHING: THE PROCESS AND EFFECT	105
P2: SEMANTIC-LINGUISTIC FEATURE VECTORS FOR SEARCH: UNSUPERVISED CONSTRUCTION AND EXPERIMENTAL VALIDATION	117
P3: RELATING ONTOLOGY AND WEB TERMINOLOGIES BY FEATURE VECTORS: UNSUPERVISED CONSTRUCTION AND EXPERIMENTAL VALIDATION	133
P4: MEASURING INTRINSIC QUALITY OF SEMANTIC SEARCH BASED ON FEATURE VECTORS.....	145
P5: CONSTRUCTING FEATURE VECTORS FOR SEARCH: INVESTIGATING INTRINSIC QUALITY IMPACT ON SEARCH PERFORMANCE.....	163
P6: AN ONTOLOGY-DRIVEN APPROACH TO WEB SEARCH: ANALYSIS OF ITS SENSITIVITY TO ONTOLOGY QUALITY AND SEARCH TASKS	183
P7: CROSS-LINGUAL INFORMATION RETRIEVAL BY FEATURE VECTORS.....	197
P8: SCENARIO-DRIVEN INFORMATION RETRIEVAL: SUPPORTING RULE-BASED MONITORING OF SUBSEA OPERATIONS	209

APPENDICES	219
A: SECONDARY PAPERS.....	221
B: EXPERIMENT INVITATION LETTER	225
C: EXPERIMENT INTRODUCTION LETTER.....	227
D: INTRODUCTION TO THE PROTOTYPE	229
E: SIMULATED INFORMATION NEEDS	233
F: QUESTIONNAIRE.....	235
G: RESULTS OF THE QUESTIONNAIRE	243
H: WORKSHOP.....	257
I: ONTOLOGIES	259

List of Figures and Tables

List of figures

FIGURE 1.1: AN ILLUSTRATION OF AN AMBIGUOUS SEARCH.....	3
FIGURE 1.2: THREE DIFFERENT KINDS OF CHRISTMAS TREE.....	4
FIGURE 1.3: THE RELATIONSHIP BETWEEN HOMONYMS AND SYNONYMS.	5
FIGURE 1.4: AN ILLUSTRATION OF A DISAMBIGUATED SEARCH.	6
FIGURE 1.5: AN OVERVIEW OF HOW THE PAPERS RELATE TO THE WORK OF THIS THESIS.	14
FIGURE 2.1: VARIABLES IN AN EXPERIMENT	18
FIGURE 2.2: AN OVERVIEW OF THE RESEARCH DESIGN.....	19
FIGURE 3.1: ASPECTS OF SEMANTIC SEARCH SYSTEMS.....	24
FIGURE 3.2: THE SYSTEM FLOW OF ONTOSEARCH	28
FIGURE 3.3: THE RELATIONSHIP BETWEEN CONCEPTS AND EXTENSIONS	29
FIGURE 3.4: ILLUSTRATION OF THE SEARCH PROCESS	29
FIGURE 3.5: THE PROPOSED APPROACH TO QUERY PROCESSING.....	30
FIGURE 3.6: THE ARCHITECTURE OF THE SEMANTIC WEB SEARCH ENGINE.....	31
FIGURE 3.7: A SCREENSHOT OF HAKIA.....	33
FIGURE 3.8: A SCREENSHOT OF POWERSSET.	34
FIGURE 3.9: A SCREENSHOT OF SENSEBOT.	35
FIGURE 3.10: A SCREENSHOT OF TRUE KNOWLEDGE.	36
FIGURE 3.11: A SCREENSHOT OF YEBOL.	37
FIGURE 3.12: RELATIONSHIPS BETWEEN TERMS AND QUERY IN DOCUMENT VECTOR SPACE	39
FIGURE 3.13: AN EXAMPLE OF GENERATING A FEATURE VECTOR.....	40
FIGURE 3.14: THE PROCESS OF CONSTRUCTING PRIMITIVE CONCEPTS.	41
FIGURE 3.15: THE TOPIC SIGNATURE CONSTRUCTION PROCESS	42
FIGURE 3.16: THE ONTOLOGICAL PROFILE CONSTRUCTION PROCESS.	42
FIGURE 3.17: THE PROPOSED SEMANTIC ENRICHMENT PROCESS	43
FIGURE 3.18: THE ARCHITECTURE OF CE AND RI	44
FIGURE 4.1: AN ILLUSTRATION OF THE RELATIONSHIP BETWEEN A FEATURE VECTOR, AN ENTITY AND A SET OF DOCUMENTS.	52
FIGURE 4.2: THE ARCHITECTURE OF THE ONTOLOGY-DRIVEN INFORMATION RETRIEVAL SYSTEM.	53
FIGURE 4.3: THE SEARCH USER INTERFACE OF PROTOTYPE I.....	54
FIGURE 4.4: AN OVERVIEW OF THE SEARCH PROCESS.....	55
FIGURE 4.5: OVERVIEW OF THE FIRST FEATURE VECTOR CONSTRUCTION ALGORITHM.....	55
FIGURE 4.6: OVERVIEW OF THE SECOND FEATURE VECTOR CONSTRUCTION ALGORITHM.....	56
FIGURE 4.7: DESIGN OF EXPERIMENT I.....	58
FIGURE 4.8: AN OVERVIEW OF THE FV CONSTRUCTION PROCESS.....	61
FIGURE 4.9: DESIGN OF EXPERIMENT III.	64
FIGURE 4.10: RELEVANCE SCORES AND CRONBACH'S ALPHA FOR SELECTED ENTITIES.	66
FIGURE 4.11: TOP 1 FV QUALITY SCORES RELATIVE TO THE LOWEST SCORE.	67
FIGURE 4.12: COMPARISON OF ONTOLOGY QUALITY AND SEARCH PERFORMANCE W.R.T. SEARCH TASKS. .	76

List of tables

TABLE 1.1: TEXT FRAGMENTS RELATED TO DIFFERENT KINDS OF CHRISTMAS TREES.	5
TABLE 1.2: RESEARCH OVERVIEW.....	12
TABLE 3.1: SUMMARY OF REVIEWED ACADEMIC APPROACHES TO SEMANTIC SEARCHES.	31
TABLE 3.2: SUMMARY OF REVIEWED COMMERCIAL APPROACHES TO SEMANTIC SEARCH.	37
TABLE 3.3: SUMMARY OF EVALUATION APPROACHES.....	47
TABLE 4.1: DEMOGRAPHIC AND BACKGROUND INFORMATION ABOUT THE PARTICIPANTS.....	59
TABLE 4.2: COMPARISON OF MEAN RELEVANCE SCORE OF KEYWORD AND CONCEPT BASED SEARCHES.	59
TABLE 4.3: AVERAGE RELEVANCE SCORES VERSUS ONTOLOGY VERSION.	59
TABLE 4.4: MEAN SCORES ON QUESTIONNAIRE ITEMS REGARDING THE EXPERIMENT.	60
TABLE 4.5: ONTOLOGY KEY CHARACTERISTICS.	62
TABLE 4.6: SUMMARY OF QUALITY PARAMETERS USED TO CONSTRUCT THE FVs.....	65
TABLE 4.7: FV QUALITY SCORES W.R.T. DIFFERENT CONSTRUCTION PARAMETERS.....	66
TABLE 5.1: PUBLISHED PAPERS ANSWERING RESEARCH QUESTIONS.	81
TABLE 5.2: RELATIONSHIPS BETWEEN THE CONTRIBUTIONS AND THE PUBLISHED PAPERS.....	83

Glossary

Class: See *Entity*.

Cluster Feature Vector (CLFV): A cluster feature vector is a *Feature Vector* that is associated with a cluster of documents.

Concept: See *Entity*.

Document Feature Vector (DFV): A document feature vector is a *Feature Vector* of a document. A document can be either a full text document, retrieved from the Web, or a snippet (i.e. a focused summary of a Web page provided by a search engine to indicate the content of the Web page).

Entity: An entity can be either a class or an individual of an ontology. We use the term entity instead of concept because a concept is often used as a synonym for a class in the Semantic Web. Since our approach constructs feature vectors for both classes and individuals, they are commonly referred to as entities in this work.

Feature Vector (FV): A feature vector is a set of key-phrases and corresponding frequencies associated with the beholder of the feature vector (i.e. concept, document and cluster). See Section 4.1 for a formal definition of feature vectors.

Feature Vector Construction (FVC): The process of constructing Feature Vectors for each entity of an ontology based on a text corpus.

Individual: See *Entity*.

Information Retrieval (IR): According to Baeza-Yates and Ribeiro-Neto (1999), "Information retrieval (IR) deals with the representation, storage, organisation of, and access to information items".

Instance: See *Entity*.

Named entity (NE): A word or a combination of words in a piece of text that is referred to by a name (i.e. organisation, people, country, location). For example, Apple as a company can be a named entity while apple as a fruit is not. Numbers are also referred to as named entities.

Ontology: An ontology is a kind of knowledge model. Ontologies can define concepts and the relationships among them for a domain of interest. According to Gruber (1993) "an ontology is an explicit specification of a conceptualization".

Ontology-based Information Retrieval (ObIR): See *Ontology-driven Information Retrieval*.

Ontology-driven Information Retrieval (OdIR): An approach to information retrieval that utilises one or more ontologies to improve the retrieval effectiveness.

Phrase: A phrase is a group of words (see *Word*) or terms (see *Term*) forming a part of a sentence.

Precision and recall: Precision and recall are the most commonly used IR (see *Information Retrieval*) metrics. Precision denotes the fraction of retrieved documents that are relevant while recall denotes the fraction of relevant documents that are retrieved (Manning et al., 2008, p. 142).

Query: A combination of one or more terms (see *Term*) normally intended to express the information need of a user. The query is submitted to a search engine that retrieves assumed relevant information.

- Retrieval effectiveness:** The retrieval effectiveness of an information retrieval system is the overall performance of a system seen as a combination of several measures like relevance (see *Precision and recall*) and user satisfaction. The most frequent and basic relevance measures are precision and recall (Manning et al., 2008, p. 142). However, in this work retrieval effectiveness is defined as the users' perceived relevance of retrieved results w.r.t. the users' queries.
- Semantic search:** Our definition of semantic search complies with the definition by Wang et al. (2008) that "semantic search supplements and improves conventional information retrieval systems on the basis of structural knowledge representation formalisms".
- Semantic Web (SW):** The Semantic Web (SW) is the "Web of data" in contrast to the classical Web that is a "Web of documents" (W3C, 2001). The vision for the SW is to enable computers to do more useful processing, and hence presentation, of the vast amount of information found on the Web. An important component of the SW is *Semantic Web Documents*.
- Semantic Web Document (SWD):** A formal description of concepts and the relationship between them represented in a document. W3C has specified several formal representation languages where the Web Ontology Language (OWL) is one of the latest recommendations from W3C.
- Term:** A term is a word (see *Word*) or a combination of words forming an expression (e.g. "Christmas tree"). Note that for example "buying a Christmas tree" is considered a phrase rather than a term.
- Word Sense Disambiguation (WSD):** The process of finding the correct meaning of words in a specific context.
- Word:** A word is a unit of language, used with other words to form a sentence. Words are typically surrounded by separators like spaces or punctuation marks.
- World Wide Web (WWW):** The World Wide Web, aka the Web, is a network of information resources that are available through the Internet. The information resources are usual textual documents written in a mark-up language like HTML (i.e. hypertext) and interlinked with references (i.e. hyperlinks).

Part I

Research Context and Results

1 Introduction

In this chapter, a synopsis of the research work conducted during my doctoral studies is provided. First, the problem is outlined, with details of the background and our motivation for solving it, and then the context of this research is presented. Next, the research questions and contributions are presented and their relations explained. The research approach is also introduced, followed by the abstracts of the main papers. Finally, the structure of this thesis is laid out.

1.1 Background and motivation

The motivation for this work comes from the acknowledgement that searching for information on the Web can be both frustrating and tedious if high quality results are desired. There are several reasons, such as the vast amount of information available on the Web that make searching increasingly difficult (Horrocks, 2007), Web spamming (Baeza-Yates, 2003; Gyongyi & Garcia-Molina, 2005; Lewandowski, 2005), low quality of information (Baeza-Yates, 2003; Lewandowski, 2005), etc. Though probably, the foremost reason is that words are ambiguous (Ding et al., 2005; Horrocks, 2007).

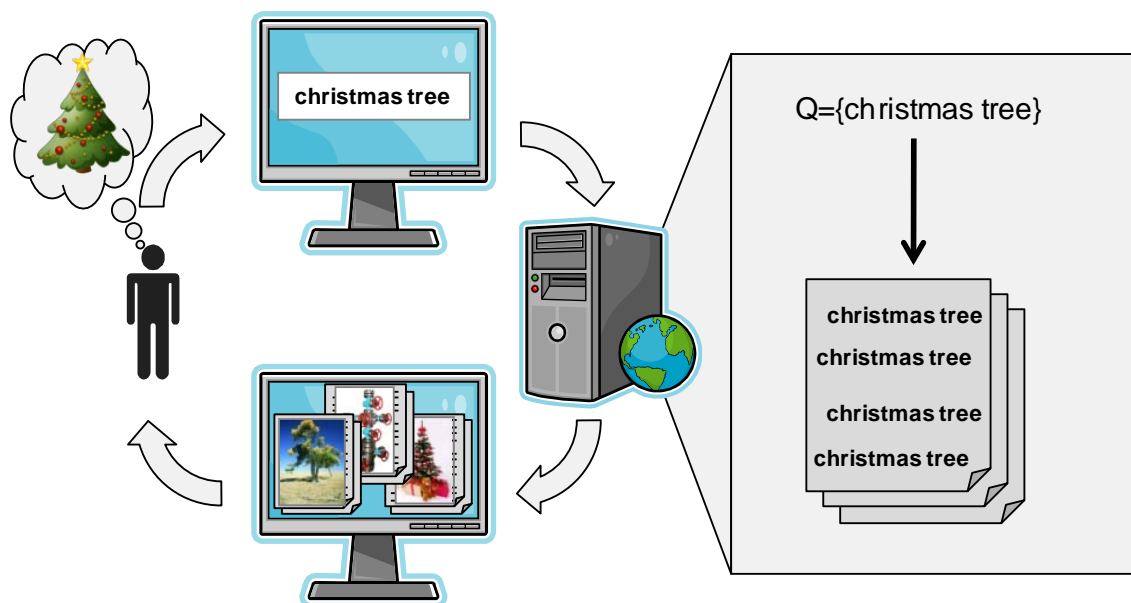


Figure 1.1: An illustration of an ambiguous search.

The problem of ambiguous words (words are hereinafter referred to as terms) in the context of information retrieval (IR) is illustrated in Figure 1.1. The user in this case, is trying to find information about *Christmas trees*. Mentally the user thinks of *Christmas tree* in the context of celebrating Christmas (i.e. a holiday held to commemorate the birth of Jesus, a central figure in Christianity). The user formulates a query that is submitted to a search engine. In a traditional search engine the query terms are matched with the terms in an inverted index consisting of all the document terms of a text corpus (Baeza-Yates & Ribeiro-Neto, 1999). Only matched documents are retrieved and

presented to the user. However, since the query in this case is ambiguous (see Figure 1.2) irrelevant results to the user's information needs are also retrieved (i.e. information about trees from Western Australia and wellheads). This little example illustrates a typical problem with ambiguous terms on the Web.

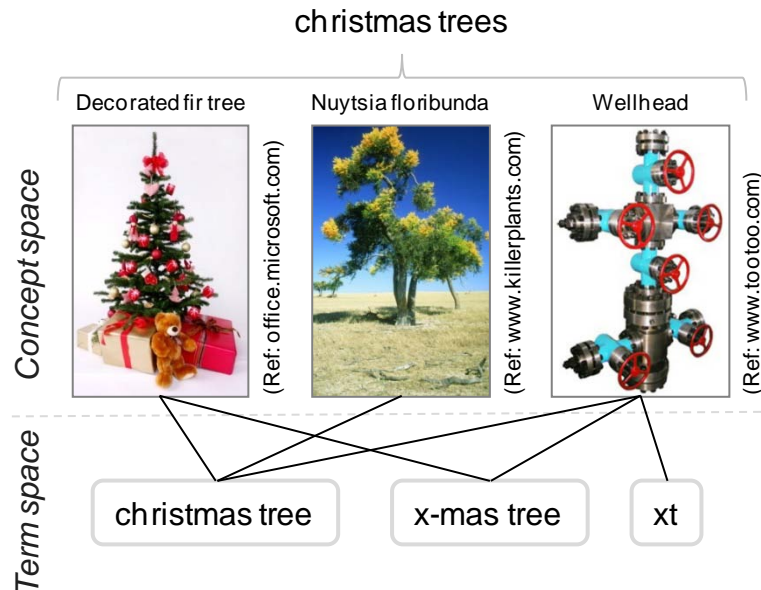


Figure 1.2: Three different kinds of *christmas tree*.

Figure 1.2 depicts an example of the term *christmas tree* used in three different domains and hence being three different concepts. *Christmas tree* is commonly associated with a decorated fir tree (see *Decorated fir tree* in Figure 1.2) when in the context of celebrating Christmas. However, *Christmas Tree* is also commonly used for a parasitic plant found in Western Australia (see *Nuytsia floribunda* in Figure 1.2) and within the oil and gas industry as a part of a wellhead (see *Wellhead* in Figure 1.2). In addition, other interpretations of the term *christmas tree* exist. That is, a term can represent different concepts depending on its domain of use. A set of concepts having the same term representation are referred to as homonyms.

Similarly, a single concept can be represented by several different terms. For example, in the standardisation report by (Standards Norway, 2004), *christmas tree* is also referred to as *x-mas tree*, *xmas tree*, *XT* and sometimes just *tree*. Figure 1.2 depicts sample concepts, terms and relations among them. Terms that represent the same concept are referred to as synonyms.

Consequently, terms are ambiguous and can be interpreted differently. Ambiguity is minimised by considering the context of terms. Disambiguating terms by their context is a fairly effortless process for humans. However, for a computer this is a rather complicated task. Humans typically work in concept space while computers work in term space (Ozcan & Aslangdogan, 2004). Concepts are defined by how they relate to other concepts. Terms, on the other hand, consist of one or more words that represent concepts (e.g. *christmas tree*). A term can represent many concepts (i.e. homonyms) while a concept can be represented by many different terms (i.e. synonyms). The relationship between homonyms and synonyms is summarised in Figure 1.3.

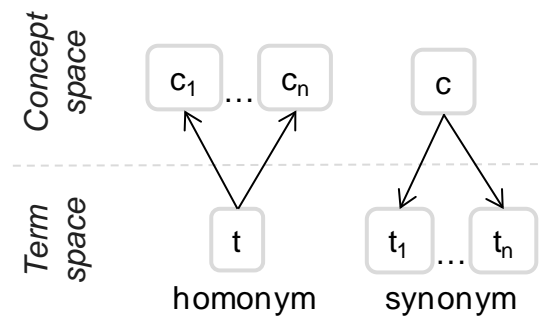


Figure 1.3: The relationship between homonyms and synonyms.

Within a domain a concept by definition possesses unambiguous meaning (e.g. in the context of celebrating Christmas a *Christmas tree* is never a wellhead). Nonetheless, individuals typically have their own connotation of concepts. For example, some think of a *Christmas tree* (see *Decorated fir tree* from Figure 1.2) with only tinsel and Christmas lights while others imagine ornaments and Christmas lights but not tinsel. In this example, there are different connotations of the common concept *Christmas tree* while the overall conceptual notion of the concept is shared. Typically within industries and disciplines the terminology is more formally defined, however different connotations are still common (Sandsmark & Mehta, 2004).

Table 1.1: Text fragments related to different kinds of *christmas trees*.

Christmas tree		
<i>Decorated fir tree</i>	<i>Nuytsia floribunda</i>	<i>Wellhead</i>
<p>“The Christmas tree is a decorated evergreen coniferous tree, real or artificial, and a tradition associated with the celebration of Christmas or the original name Yule.”</p> <p>(Ref: en.wikipedia.org)</p>	<p>“The moodjar (<i>Nuytsia floribunda</i> (Labill.) R. Brown) of Western Australia is a hemiparasite, a mistletoe. Unlike other mistletoes in its family, the Loranthaceae, the moodjar does not grow upon the above-ground portions of host plants. Nor does it remain shrubby. It is the largest of the mistletoes, growing to 10 meters (30 feet).”</p> <p>(Ref: www.killerplants.com)</p>	<p>“In petroleum and natural gas extraction, a Christmas tree, or “Tree”, (not “Wellhead” as sometimes incorrectly referred to) is an assembly of valves, spools, and fittings, used for an oil well, gas well, water injection well, water disposal well, gas injection well, condensate well and other types of wells.”</p> <p>(Ref: en.wikipedia.org)</p>
<p>“The fir tree has a long association with Christianity, it began in Germany almost 1,000 years ago when St Boniface, who converted the German people to Christianity, was said to have come across a group of pagans worshipping an oak tree.”</p> <p>(Ref: www.christmas-tree.com)</p>	<p>“Nuytsia floribunda is a parasitic plant found in Western Australia. The species is known locally as the Christmas Tree, displaying bright orange flowers during the Christmas season.”</p> <p>(Ref: en.wikipedia.org)</p>	<p>“An assembly of valves, spools, pressure gauges and chokes fitted to the wellhead of a completed well to control production. Christmas trees are available in a wide range of sizes and configurations, such as low- or high-pressure capacity and single- or multiple-completion capacity.”</p> <p>(Ref: www.glossary.oilfield.slb.com)</p>

In Table 1.1, two text fragments for each of the three *christmas tree* concepts are shown. In this example, the emphasis in the text fragments is added manually. As can be seen from the example different words are used to describe the shared concepts, though some of the terms for each domain are common. Since each individual uses

different words when describing common concepts in documents it can be difficult to retrieve those documents. For example, in the text fragment from *en.wikipedia.org* regarding the *Decorated fir tree* (see Table 1.1), the term *evergreen coniferous tree* is used to describe the Christmas tree concept. While in the text fragment from *www.christmas-tree.com* the term *fir tree* is used. Consequently, a user searching for *evergreen coniferous tree* will not necessarily get results from *www.christmas-tree.com* even if the term *Christmas tree* is part of the query.

Bear in mind that the motivation for this work came from the acknowledgment that finding highly relevant information on the Web can be both frustrating and tedious. In Figure 1.1, an illustration of an ambiguous search was provided, while in Figure 1.4 an updated example is provided with a system capable of disambiguating search. In this scenario, the information need is the same as in the previous example. However, in this case the search engine is concept based. The search engine being concept based means that it works on a semantic level (i.e. concept space) instead of on a lexical level (i.e. term space). Idealistically, the concept based search engine is capable of disambiguating search and hence retrieves results that better fit the information needs of the user.

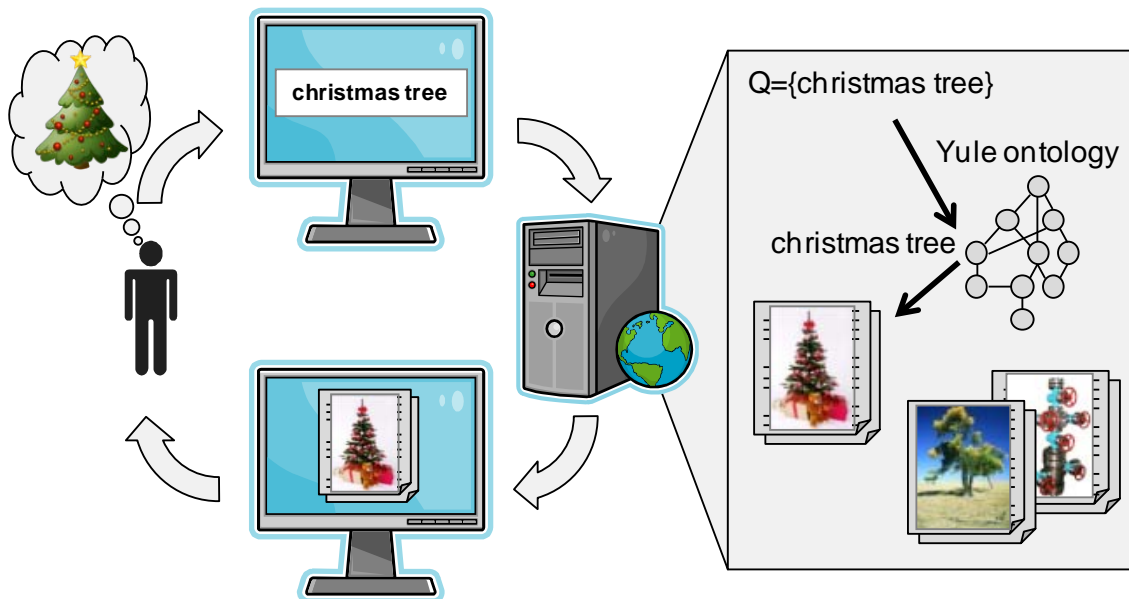


Figure 1.4: An illustration of a disambiguated search.

In this thesis, we explore alternative approaches to semantic annotation and word sense disambiguation for the Web. In the example depicted in Figure 1.4, an ontology was used to disambiguate search. We explore how the mapping of concepts in ontologies to terminologies used in textual documents of a corpus (i.e. the Web) can be done in a flexible manner (i.e. to avoid static linking between concepts and documents).

As a starting point, this work builds upon the following prerequisites/assumptions:

- Web search using standard query language
- Multitude of ontologies are available
- Documents not being ontologically annotated

1.2 Problem outline

In computer science, the process of mapping terms to concept-space is referred to as Word Sense Disambiguation (WSD). In general there are two main approaches to WSD; supervised and unsupervised (Navigli, 2009). Supervised approaches typically use machine-learning techniques that learn to classify senses from examples (i.e. training sets). In contrast, the unsupervised approaches do not depend on training sets but instead use techniques to utilise the information provided in the applied corpus (e.g. word collocation, keywords and part-of-speech). These approaches can further be divided into knowledge-based and knowledge-poor approaches (Navigli, 2009). The former approaches use external knowledge resources (i.e. knowledge model, thesaurus, taxonomy) while the latter do not depend on external resources (i.e. statistics).

The basic problem in WSD is the mapping from term-space (i.e. documents) to concept-space (i.e. knowledge models). Ontologies are one form of knowledge model that formally represent a universe of discourse by describing the relationships between its concepts (Gruber, 1993). Enriching documents with machine understandable mark-up (i.e. metadata) is one knowledge-based approach. The aim of extending documents with metadata is, among others, to remove ambiguities in the documents. The metadata is either descriptive data about the document as whole (i.e. document level) or about terms in the document (i.e. named entities, concepts). Furthermore, the metadata can either be embedded into the document or stored separately. Metadata at a document level normally includes general data like authors, keywords, etc. (Kobayashi & Takeda, 2000), while on the term level it typically includes references to entities in an ontology (Desmontils & Jacquin, 2001; Kiryakov et al., 2004; Lopez et al., 2006b; Popov et al., 2003). Metadata at the document level is hardly used by any search engines when indexing documents since it can be and has been misused for the purpose of giving the documents a misleading higher ranking than it should have (Kobayashi & Takeda, 2000; Sullivan, 2002). Manually annotating Web documents using knowledge models can be tedious, labour-intensive and error prone, and consequently not practical for real life applications (Reeve & Han, 2005; Rehbein et al., 2009). Consequently, most approaches do this either automatically or semi-automatically (Escudero et al., 2000; Kiryakov et al., 2004).

Approaches to semantic annotation mainly focus on either using a domain ontology or a small set of ontologies (Kiryakov et al., 2004; Laclavik et al., 2007). A WSD application targeting the Web must be able to handle millions of ontologies. Consequently, a concern with these semantic annotation platforms is their ability to cope with these numbers of ontologies. Currently, according to Swoogle (Swoogle, 2005) a Semantic Web Document (SWD) search engine for the Web, there are more than 3 million SWDs (i.e. ontologies) available on the Web. The number of SWDs will continue to grow.

Most semantic annotation approaches create mappings between the documents and the ontologies (i.e. typically an annotated term or document becomes an instance of an ontology) (Reeve & Han, 2005; Uren et al., 2006). This can be an ideal solution when retrieving documents. However, from a maintenance perspective this approach becomes increasingly difficult to apply with billions of documents that are mapped to millions of concepts in various ontologies (Kiryakov et al., 2004; Uren et al., 2006). Reasoning

over ontologies will also be increasingly difficult due to the sheer scale of the task (Ding et al., 2005). Therefore, an alternative to traditional semantic annotation approaches being flexible with respect to the Web needs to be explored.

In this work, we focus on how to enhance search performance by exploiting semantics defined in ontologies. Ontologies are built for various purposes, but all of them express perceived relations between concepts, and consequently they can provide useful models for information retrieval purposes. Therefore, we will explore the possibilities to connect domain terminology (encoded in ontologies) to the actual terminology used on the Web (recall Table 1.1). The underlying idea is that terms in a particular domain can be associated with ontology concepts that reflect both the semantic and linguistic neighbourhoods of the concepts. The semantic neighbourhood can be computed based on related concepts and direct properties specified in an ontology, while the linguistic neighbourhood can be based on collocations of terms in a text corpus like the Web, e.g. expressed by weights using the Vector Space Model (Manning et al., 2008, p. 110). We aim to develop an unsupervised (i.e. independent of an already semantically annotated text corpus) and knowledge-based (i.e. use of ontologies) approach that is robust with respect to the Web. However, the result can vary a lot depending on the quality of ontologies. Therefore, we will also explore aspects of ontologies that can influence search effectiveness (i.e. ontologies of different granularity like taxonomy versus more advanced ontologies, etc.).

Evaluation of information retrieval systems concerns assessing their retrieval efficacy - delivering more relevant information faster. That is, they are evaluated with respect to improved efficiency - the system response time, user interaction time, etc. In addition their effectiveness is evaluated with respect to recall and precision - more relevant results. Since the focus of this thesis is to improve existing Web search systems (implying the addition of a component on top of current Web search engines), it will result in increased (however insignificant) interaction and response time. Therefore, our focus is to improve effectiveness, specifically looking at quality of the retrieval rather than the optimisation of other parameters like space usage (i.e. index size). Moreover, there are distinguished two main stream approaches to evaluate effectiveness: system- and user-centric. System-centric evaluation is the most common and typically uses traditional basic relevance measures like precision and recall (Manning et al., 2008, p. 142). Relevance is normally assessed by human-judgment or by using a standard document collection like TREC (Voorhees & Harman, 2005). However, Harter (1996) argues that using a fixed set of documents and queries does not reflect reality. In addition, it is widely accepted that external factors exist that can considerably affect the retrieval results like query quality and familiarity of search topic (Alemayehu, 2003; Gao et al., 2004; Harter, 1996). User-centric approaches evaluate users' satisfaction by viewing the system as a whole and involving the users. Sometimes user satisfaction is equated with system effectiveness (Su, 1992). However, user-centric approaches are less scalable and repeatable than system-centric approaches (Huffman & Hochster, 2007). Since the ultimate goal of any IR evaluation is to assess the probability of an IR system being both adopted and used, potential end-users must be involved. Therefore, by retrieval effectiveness we mean users' perceived relevance of retrieved results w.r.t. the users' queries. That is, we seek to enhance precision of results without adding significant complexity on user interaction.

1.3 Research context

The research in this thesis is part of the Integrated Information Platform for Reservoir and Subsea Production Systems (IIP) project (Sandsmark & Mehta, 2004) supported by the Norwegian Research Council (NFR). The NFR project number is 163457/S30. The project started in 2004 and was finished in 2007.

The goal of the IIP project is to extend and formalise existing terminology for the petroleum industry standard ISO 15926 (Gulla et al., 2006). ISO 15926 consists of seven parts, but part 4 (ISO, 2007), the Reference Data Library (RDL), is the focus of the project. Part 4 is comprised of application and discipline-specific terminologies but the project focuses on terminologies for subsea equipment used by the oil and gas industry in particular. These terminologies, described as RDL classes, are instances of the data types from part 2. Part 2 defines the language for describing standardised terminologies, while part 4 describes the semantics of these terminologies. An objective is to define an unambiguous terminology of the domain and build an ontology that will ease the integration of systems between disciplines. A common terminology is assumed to reduce risks and improve the decision making process in this industry.

The success of this new ontology, and standardisation work in general, depends on the users' willingness to commit to the standard and devote the necessary resources (Gulla et al., 2006). If people do not find it worthwhile to take the effort to follow the new terminology, it will be difficult to develop the necessary support. Therefore, intelligent ontology-driven applications must demonstrate the benefits of the new technology and convince the users that the additional sophistication pays off (Strasunskas & Tomasgard, 2010).

Further, creating and maintaining ontologies is both time-consuming and costly (Simperl et al., 2009). Consequently, ontologies ought to be applied for as many different tasks as possible to increase the return on the investment. Therefore, another focus of the IIP project is reuse of the created ontology for rule-based notification and semantic search (Gulla et al., 2006). Part 4 of the ISO 15926 ontology (ISO, 2007) will also be specified in the Web Ontology Language (OWL). Therefore, the project seeks to apply this ontology to the semantic search application created as part of the research conducted in this thesis (see Section 1.5 for more information). Considering multi-disciplinary domains and a big variation in terminology used in the oil and gas industry, one of the challenges is adaption of the ontology to a document space (text corpus).

Given the amount of existing search systems, the semantic search approach ought to be applied on top of these existing systems, extending them with semantic capabilities. That is, the indexing and ranking components of the systems ought to be unaltered while the query and presentation components can be extended with semantic search techniques (i.e. use of ontologies). The semantic search approach should be able to disambiguate queries by utilising the knowledge provided in ontologies.

1.4 Objectives and research questions

Based on the principles discussed in Section 1.3, the main objective of this research was formulated as follows:

MO: *Improve information retrieval effectiveness by means of ontologies.*

Improve the effectiveness (see Section 1.2) of an information retrieval system by utilising ontologies. Ontologies describe how concepts relate to other concepts within particular domains, therefore utilise these relationships to improve information retrieval effectiveness.

This main objective was split into the following two sub-objectives:

SO1: *Explore and analyse approaches to connecting the domain terminology provided in ontologies to the actual terminology provided in textual documents.*

Recall from Section 1.2 that textual documents can be annotated with metadata that can be utilised to perform word sense disambiguation. The objective is to explore in literature and analyse alternative approaches for associating terminologies found in ontologies with terminologies used in text corpora.

SO2: *Develop an effective method for applying ontologies to existing search systems.*

While the objective of SO1 is to explore and analyse approaches of connecting terminologies found in ontologies and text corpora. The objective here is to develop an effective method for connecting the terminologies. The method must be applicable to existing search systems by extending the typical query and presentation components without altering the indexing and ranking components of these systems.

Based on the objectives above a set of research questions was formulated as follows:

RQ1: *Can the retrieval effectiveness of search systems be improved by utilising ontologies?*

Determine in the literature whether the effectiveness of information retrieval (i.e. quality of search results) can be improved by utilising ontologies (see also RQ4). Can ontologies be used to handle ambiguity in search queries (recall MO)? Can ontology concepts be related to terms in documents and queries (recall SO1)?

RQ2: *How can the terminology provided in an ontology be related to terms in textual documents and queries?*

Develop an effective method for connecting terminologies in ontologies with terminologies used in text corpora. How can this method be applied to existing search systems (recall SO2) and extend these systems with semantic technology techniques (i.e. ontologies)?

RQ3: *How can the quality of the associations between the concepts of an ontology and a text corpus be evaluated?*

Explore and develop a method for evaluation of the quality of association between the concepts of an ontology and related terms in a text corpus.

RQ4: *What features of an ontology influence the search performance?*

Explore aspects of the ontologies that can influence the search effectiveness. Find to what extent the approach is sensitive to the quality of the ontologies. To what extent is the approach indifferent for ontologies of different types (i.e. different granularity/quality)? To what extent is the approach independent of the processing sequence of the ontology concepts?

1.5 Research approach and scope

In this section, we provide an overview of the research conducted as part of this work.

The research method applied to this work can be classified as problem-solving research (Phillips & Pugh, 2005). The work was divided into three phases: (1) Analysis and design, (2) Prototype I, and (3) Prototype II. The phases were conducted in a consecutive order. Experiences and results from earlier phases influenced the work of the next phase. Each phase addressed at least one research question and resulted in one or more contribution (Table 1.2). The research phases are:

Phase I: Analysis and design

The objective of this phase was to formulate a set of theories for this work. Therefore, a broad literature study was conducted to get an understanding of the current state-of-the-art. Based on the acquired understanding, a set of theories was formulated and partly tested. This work and the lessons learned formed the foundation for the design of the approach.

Phase II: Prototype I

The objective was to validate the set of theories formulated in the previous phase. A prototype was implemented in Java to validate the feasibility of the proposed approach. A set of experiments was designed and conducted with real, and potential future, users of such a system. The lessons learned from the proposed approach in this phase influenced the formulation of new theories to be validated.

Phase III: Prototype II

The objective of this phase was to get a better understanding of the proposed approach. Therefore, a new prototype was implemented, reflecting the lessons learned from the first prototype (i.e. new algorithms). New experiments were designed with a focus on aspects of the main components of the proposed approach. One of the goals of the experiments was to get a better understanding of the sensitivity of the approach and how this approach could be evaluated in an effective manner.

Furthermore, each research phase included a cycle of four tasks. They are discussed in detail in Chapter 2.

As mentioned in Section 1.3, the intention was that Part 4 of the ISO 15926 (ISO, 2007) ontology covering subsea equipment, would be applied to the semantic search application created as part of this work. However, as it turned out it was impossible to get access to a text corpus being both big enough and related to this ontology. Therefore, we were not able to construct feature vectors (see Glossary) (i.e. there needs

to be a correlation between the ontology and the documents) and hence were not able to test the suitability of this ontology for searching. Instead, another set of ontologies was selected. They were supposed to cover topics of interest to test search applications (i.e. ambiguous terms that are commonly used and hence can be a challenge for common search engines), and the topics should be commonly known - that is rare topics should be avoided. Furthermore, the ontologies should be of different types and ideally used in other research projects. We chose to exclude ontologies with several thousand entities since they were not believed to provide any significantly new insight except that of processing time, which is not a focus of this work. Based on these criteria a set of ontologies (see Appendix I) was selected and used throughout the experiments conducted as part of this work.

Table 1.2: Research overview.

	Research phases		
	<i>Phase I: Analysis and design</i>	<i>Phase II: Prototype I</i>	<i>Phase III: Prototype II</i>
<i>Research questions</i>	RQ1, RQ2	RQ1, RQ2	RQ3, RQ4
<i>Contributions</i>	C1, C2	C1, C3	C4, C5
<i>Research methods</i>	Literature study and controlled experiment	Controlled experiment and questionnaire	Controlled experiment
<i>Publications</i>	P7, P8	P1, P6	P2, P3, P4, P5

As mentioned, two prototypes were implemented in Java™ and run on an Apache Tomcat® (i.e. a Java Servlet runtime environment). We used several search engines as our underlying search engine. The implementation supported Apache Nutch®, Yahoo!® and Google®. Adding support for a new search engine took about two to three hours. More information about the implementations is provided in Section 4.2.

1.6 Contributions

The research work was conducted in three phases as shown in Table 1.2, where each phase provided a set of results. The results, described as contributions of this work, have been published in peer-reviewed international conferences and journals. In addition, an international workshop on "Aspects in Evaluating Holistic Quality of Ontology-based Information Retrieval" (ENQOIR) was organised in 2009 (see Appendix G).

The contributions of this work are summarised as follows:

C1: *An approach to improving the effectiveness of existing Web search systems by means of ontologies.*

In paper P6, we showed how the proposed approach can extend existing Web search systems with semantic techniques using ontologies. The core components (i.e. indexing and ranking) are unaltered, while the query and presentation components of these systems can be altered to support the use of feature vectors (FVs) and hence ontologies to improve their effectiveness. A FV connects an entity to the specific terminology used in a particular document collection and

constitutes a rich representation of an entity by containing the actual terminology both associated and used in the document collection.

- C2:** *A flexible approach applicable to multilingual and task-driven search applications.*

The proposed approach of feature vectors (FVs) can be applied to a variety of different search applications. In this thesis, we have explored the use of FVs in three different search applications. In paper P7, we proposed a cross-lingual information retrieval approach where a set of query terms with related concepts is translated into a different language by utilising FVs. While in paper P8, we proposed a scenario-driven information retrieval approach to improve task related information retrieval that required the tailoring of FVs to provide increased quality of search results. Third, and the main approach (paper P6), was a proposed Web search application utilising FVs to disambiguate search and hence improve the precision of the search results.

- C3:** *An unsupervised approach to associate entities from ontologies with related terminologies in textual documents.*

In paper P1, we proposed an approach where every ontology entity is associated with a feature vector tailored to the specific terminology of a text corpus. This unsupervised solution is applicable to any ontology and text corpus as long as there is a correlation between them. An advantage of the approach is that a diverse corpus, like the Web, can be used since our approach is capable of disambiguating word senses by utilising the relationships between the entities within an ontology.

- C4:** *A set of guidelines and parameters for optimising feature vectors with respect to ontology quality.*

Conducted experiments (paper P2 and P3) let us empirically derive a set of guidelines and parameters on how to construct optimal feature vectors. These guidelines and parameters are optimal with respect to both ontology quality and ontology granularity.

- C5:** *An evaluation framework for assessing feature vectors' quality with respect to both the ontology and the text corpus used.*

In paper P4, we proposed a framework that uses both intrinsic and extrinsic measures to evaluate the quality of the associations. The intrinsic measure evaluates the associations with respect to the ontology used, while the extrinsic measure utilises the vast amount of information found on the Web to perform the evaluation. In addition, since the Web is constantly changing, a measure to account for the drifting effect of the Web was proposed. In paper P5, we validated this evaluation framework with real users.

An overview of the contributions and how they relate to the published papers and the research phases is shown in Table 1.2 and Figure 1.5.

1.7 Overview of main publications

As part of this work, eight main papers have been published in peer-reviewed international conferences and journals. In this section, we provide a list of the publication details of these papers. The papers, P1-P8, are summarised in Chapter 4 and included in Part II of this thesis. An overview of the papers and their relationship to the rest of this work is shown in Figure 1.5.

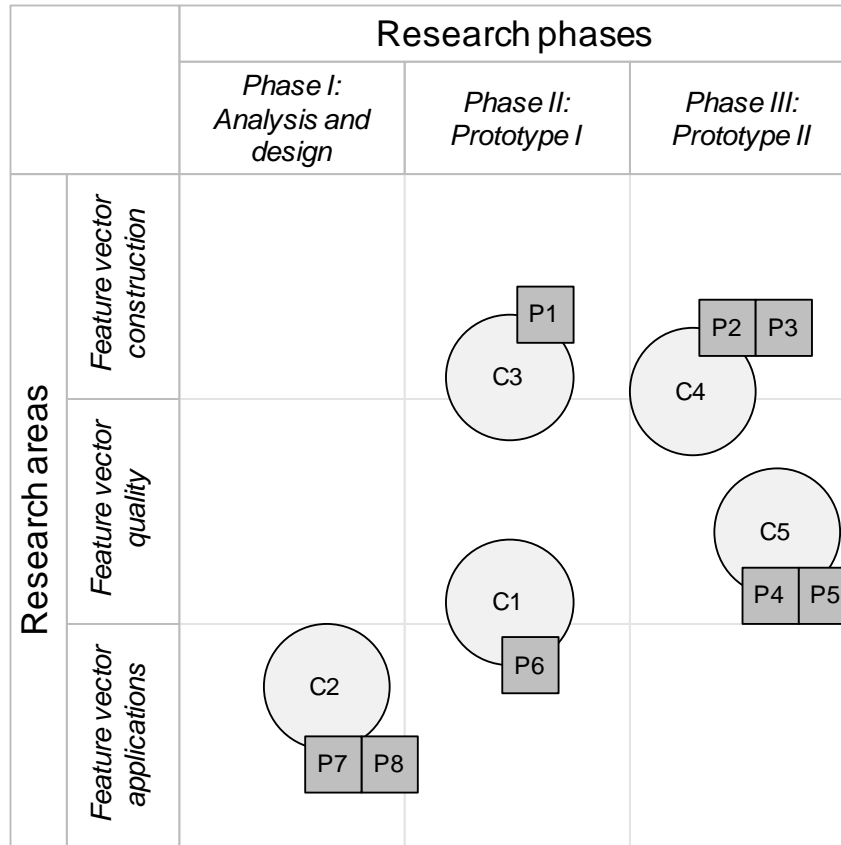


Figure 1.5: An overview of how the papers relate to the work of this thesis.

- P1:** Tomassen, S.L. & Strasunskas, D. (2009) Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect. In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, IEEE Computer Society, Washington, pp. 133-138.
- P2:** Tomassen, S.L. & Strasunskas, D. (2009) Semantic-Linguistic Feature Vectors for Search: Unsupervised Construction and Experimental Validation. In: Gomez-Perez, A., Yu, Y. & Ding, Y. (eds.) *The Semantic Web*, LNCS 5926, Springer, Heidelberg, pp. 199-215.
- P3:** Tomassen, S.L. & Strasunskas, D. (2009) Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation. In: Kotsis, G., Taniar, D., Pardede, E. & Khalil, I. (eds.) *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, ACM, pp. 86-93.

- P4:** Tomassen, S.L. & Strasunskas, D. (2010) Measuring intrinsic quality of semantic search based on Feature Vectors. *Int. J. Metadata, Semantics and Ontologies*, 5(2), pp. 120-133.
- P5:** Tomassen, S.L. & Strasunskas, D. (2010) Constructing Feature Vectors for search: investigating intrinsic quality impact on search performance. *Int. J. Web and Grid Services*, 6(3), pp. 289-312.
- P6:** Tomassen, S.L. & Strasunskas, D. (2009) An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks. In: Kotsis, G., Taniar, D., Pardede, E. & Khalil, I. (eds.) *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, ACM, pp. 128-136.
- P7:** Lilleng, J. & Tomassen, S.L. (2007) Cross-Lingual Information Retrieval by Feature Vectors. In: Kedad, Z., Lammari, N., Metais, E., Meziane, F. & Rezgui, Y. (eds.) *Natural Language Processing and Information Systems*, LNCS 4592, Springer, Heidelberg, pp. 229-239.
- P8:** Strasunskas, D. & Tomassen, S.L. (2007) Scenario-Driven Information Retrieval: Supporting Rule-Based Monitoring of Subsea Operations. *Information Technology and Control*, 36(1A), pp. 87-92.

In addition, this research has contributed with other publications that are not included in this thesis. These secondary papers are listed with publications details in Appendix A.

1.8 Thesis structure

This thesis is divided into two parts:

- Part I:** The remainder of Part I includes a summary of related work, research approach, results and evaluation. Part I is finishes with conclusions and directions for further work.
- Part II:** Contains the papers P1-P8 listed above. The papers provide detailed descriptions of the activities and results summarised in Part I.

In more detail, Part I consists of the following chapters:

- Chapter 2 - Research Approach:** In this chapter we present the research approach used, the research phases and tasks. Furthermore, we describe some of the research methods used.
- Chapter 3 - Related Work:** This chapter provides an overview of related work. We focus on approaches using Semantic Web techniques for the enhancement of searching and construction of feature vectors, with a particular focus on the latter.
- Chapter 4 - Results:** This chapter presents the results of this work. We provide an overview of the results published in the papers presented in Part II.
- Chapter 5 - Evaluation:** Here we evaluate the results of this work presented in chapter 4. We revisit the objectives and the research questions. We evaluate the research questions with regard to the published results and hence the contributions of this work.
- Chapter 6 - Conclusions and Future Work:** Finally, in this chapter we conclude this work and propose some future research directions.

The references are found at the end of Part I, while the appendixes are provided at the end of this thesis. The appendixes include an overview of secondary papers, details about the experiments (invitation letter, simulated information needs, questionnaire, etc.), information about a workshop held, and an overview of the ontologies used in the experiments.

2 Research Approach

In this chapter, the overall research approach is presented and discussed. First, we introduce a general classification of research approaches. Then, we describe the chosen research approach and the empirical methods used in this work.

2.1 Introduction

Traditionally, research has been classified as two types: *pure-* and *applied-research*. *Pure research* deals with theories while *applied research* deals with testing of theories in the real world. However, according to (Phillips & Pugh, 2005) this twofold classification is too restrictive, i.e. it does not very well reflect the research applied in academia. Therefore, they have proposed a classification of research in three types: exploratory, testing-out, and problem solving. These classifications cover both qualitative and quantitative research methods:

Exploratory research involves research about a topic or problem about which little is known. Consequently, at an early stage of the research it can be difficult to formulate or well define the research ideas. Therefore, many different research methods may be needed or even new methods created if none is suitable. Obviously, the uncertainty can be high in such projects.

Testing-out research involves finding limitations of previously proposed generalisations. Typically, different methodologies are used to those proposed to find new insights. Alternatively, comparable methodologies are applied to get a new insight into which methodology is most suitable. Nevertheless, new insights into the previously proposed approach may be found from the experiments conducted.

Problem-solving research involves using a problem in the real world as a starting point. First, the problem needs to be defined and a methodology needs to be selected to find the solution to the problem. The process may be iterative, as it may be needed to identify new problems and hence select a new methodology as the research progresses. Real world problems tend to be complicated, therefore several disciplines may be needed to solve the problem.

The research work conducted in this thesis could be best classified as a *problem-solving* kind of research. The problem was defined from a real world setting – that is, how can ontologies be utilised to improve the effectiveness of information retrieval systems in a flexible manner (Section 1.3). Therefore, different approaches were selected to tackle the defined problem. The research approach and the methodologies are described in the following sections.

2.2 Empirical research methods

In this section, we introduce the empirical research methods used in this thesis. First, the method for controlled experiments is presented and finally questionnaires are discussed.

2.2.1 Controlled experiment

The aim of controlled experiments is to measure the effect that a set of input variables (i.e. independent variables) has on a set of output variables (i.e. dependent variables) (Wohlin et al., 2003). In addition to the independent variables, there can be external factors (i.e. confounding factors) that also can affect the dependent variables. Consequently, it is vital to identify all the confounding factors to ensure the validity of the experiment. A model with the variables of a controlled experiment is shown in Figure 2.1. Another important principle is randomization (e.g. the treatments to evaluate are distributed to the participants by random). A potential drawback is that the scope can be smaller. Consequently, these kinds of experiments require careful planning. Controlled experiments are in general suitable in cases where the relationship between variables is to be explored (e.g. for choosing best of different techniques, methods).

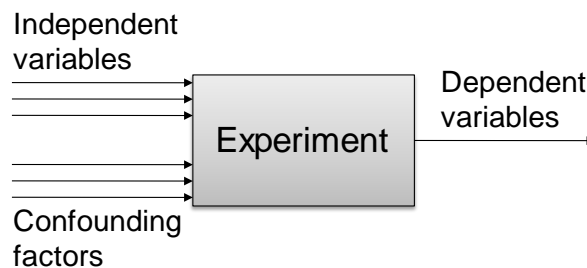


Figure 2.1: Variables in an experiment (adopted from (Wohlin et al., 2003)).

A set of standard designs for controlled experiments exist (Wohlin et al., 2003). The most basic design, and hence providing the best control over the experiment, includes using just one independent variable with only two possible values (e.g. testing a query expansion approach using two different techniques). In general, a good controlled experimental design ought to have as few independent variables and values as possible. Another issue regarding the validity of controlled experiments is that the method has been criticised for its lack of realism (Sjøberg et al., 2003). We discuss the validity of the conducted experiments in Chapter 5.

2.2.2 Questionnaire

Questionnaires or surveys are commonly used to gather data about the subjects participating in an experiment (Passmore et al., 2002). Questionnaires can be used on their own to collect data for the experiment but are typically used in conjunction with other data collecting approaches. In the latter case, a survey is used to gather data that cannot be assessed by other means and can be used to validate the other collected data.

Surveys need good planning and design in order to get a useful insight. For example, a poorly designed survey (i.e. vague questions) can provide results with a high degree of noise (i.e. inconsistent results). Nevertheless, other factors that are harder to control can influence the results. For example, the subjects can be influenced by external aspects (e.g. honesty and memory of the subjects) with respect to the questionnaire that can bias the results (Passmore et al., 2002). Therefore, the planner of a survey needs to be aware of issues that can influence on the results of the survey.

There are basically two types of survey: descriptive and explanatory (Passmore et al., 2002). Descriptive surveys capture factual data or opinions. Factual data can be gender, age, number of searches per day, etc. while opinions can be the preferred search engine, best organisation of search results, etc. Explanatory surveys attempt to capture cause and effect links (e.g. whether highlighting the query terms in a search result improves or worsens the search experience). Typically, surveys are both descriptive and explanatory.

2.3 Overall research approach

The research work in this thesis was divided into three phases; Analysis and design, Prototype I, and Prototype II (depicted in Figure 2.2). The phases were executed in a consecutive order. Lessons learned from the research conducted in the first phase influenced the work in the second phase, etc. The objective of the first phase was to get an overview of the current state-of-the-art constituting a basis for ideas. In the second phase, the objective was to test those ideas and validate theories created in the first phase by implementing a prototype and conducting an experiment with real users. The objective of the third, and last phase, was to get further insight on the construction of FVs. Therefore, we implemented a new FV construction (FVC) algorithm as part of the second prototype. The FVC algorithms were validated by conducting controlled laboratory experiments. For each of these phases a set of four tasks (i.e. a research cycle) was executed in a consecutive order (depicted in Figure 2.2). The theoretical framework is then revised for each new loop of the research cycle. The revision being based on lessons learned from the previous loop of the cycle.

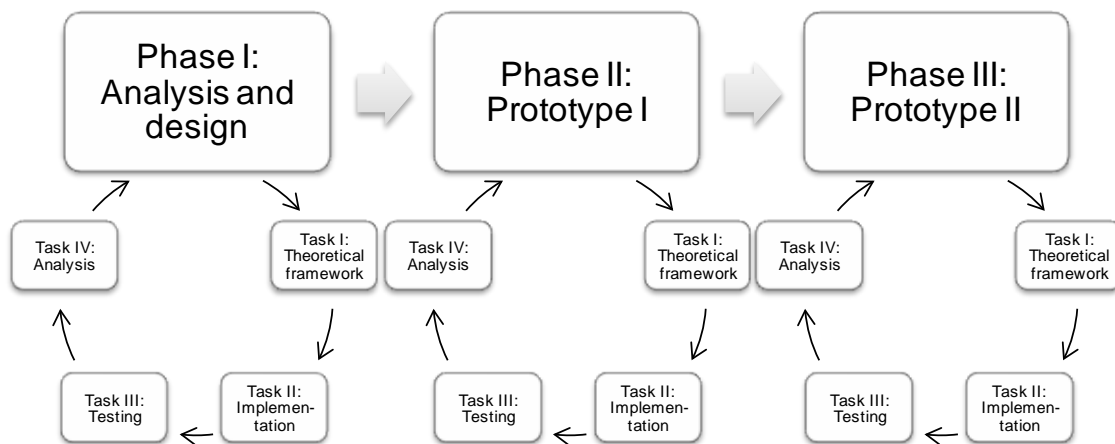


Figure 2.2: An overview of the research design.

2.3.1 Research tasks

Each of the three phases presented here includes a set of tasks conducted in a consecutive order (depicted in Figure 2.2). The four tasks are described in detail as follows.

Task I: Theoretical framework

The purpose of this task is to establish a theoretical framework functioning as a basis for Task II. This task mainly consists of conducting a literature review and establishing the

state-of-the-art within the relevant areas of this research. A new theory is created that is inspired by the literature survey and the results from the preliminary evaluations.

Task II: Implementation

The purpose of this task is to implement the theoretical framework created in Task I and prepare for the testing to be conducted in Task III. The implementation is based on the theoretical framework and a result of this is typically an application or component created in Java (more information regarding these prototypes is in Section 4.2). Other results can be a survey such as the one created in Phase II.

Task III: Testing

The purpose of this task is to test the implementation done in the previous task. The selected research method is dependent on the task. For example, for the user experiments (see Experiment I and III in Section 4.3), we adopted the measure from (Brasethvik, 2004) to obtain the perceived relevance of the search results by the users. In addition, a questionnaire was used in Experiment I since the measure by (Brasethvik, 2004) does not take into account aspects like user experience. For the laboratory experiments (Experiment II and III, Section 4.3), intrinsic and extrinsic measures were used to evaluate the quality of the feature vectors with respect to the ontologies used.

Task IV: Analysis

The purpose of this task was to analyse the results from the test conducted in Task III. The results were analysed and compared with previously gathered results. Based on this analysis the theoretical framework was revised, or a new one was created, which was then implemented, tested, etc.

2.3.2 Research phases

The research work was mainly divided into three phases that were performed consecutively (see Figure 2.2). Parallel to these phases, additional work was done that led to an international workshop being held (see Appendix H) and international publications (see list of secondary papers in Appendix A). The three phases are described in more detail as follows.

Phase I: Analysis and design

The objective of this phase was to get an understanding of the current state-of-the-art to formulate a set of theories for this work. Therefore, a broad literature study in this field of research was conducted. The understating of the current state-of-the-art and the settings discussed in Section 1.3 constituted a basis for a set of preliminary research questions and theories.

The research methods used in this phase were literature review, engineering, and experimentation. The literature review was conducted in relevant research fields. The review process was iterative; findings in one research field led to the exploration of new fields, etc. The knowledge gathered from this review led to a set of theories and overall architecture of the semantic search application. A selected set of theories was tested by experimentation. The experiments included prototyping the most vital components of

the overall system. The components were validated by testing in a controlled environment. The results from these tests were analysed. The results from the analysis affected the planning and execution of the next phase.

Phase II: Prototype I

The objective of this phase was to validate the set of theories formulated in the previous phase. To validate the theories a prototype was implemented that was later tested by real users. The user experiment included interacting with a prototype and answering a questionnaire.

The research methods used in this phase were mainly engineering, experimentation and survey research (Passmore et al., 2002). The overall architecture, engineered in the previous phase, was implemented in Java and run on a Tomcat server with a Web user interface. However, minor adjustments to the architecture were done as the implementation proceeded. The changes were done based on the testing and evaluation of system components. The prototype was tested with real users. In addition, the users were required to answer a questionnaire (see Appendix F). The objective of the survey was to acquire other kinds of information which were impossible to obtain by evaluation of the results from the prototype experiment alone.

For this experiment, the standard information retrieval metrics, precision and recall, (Baeza-Yates & Ribeiro-Neto, 1999) could be used. However, precision and recall are not well suited for the Web (Piwowarski et al., 2007). First, the relevance is coarse - it is either relevant or irrelevant, which is not the case in real life. Second, it requires the knowledge of both relevant and non-relevant documents, which is not feasible for the Web. Consequently, alternative metrics suitable for the Web were sought for (Brasethvik, 2004; Piwowarski et al., 2007; Vaughan, 2004). In this experiment we chose to adopt the measure described by Brasethvik (2004) to obtain the relevance of the search results perceived by the users. The top 10 retrieved documents were marked according to their perceived relevance (i.e. trash, non-relevant or duplicate, related, or good) and weighted according to their ranked positions. This gives a final score in the range [-50, 100]. This score substitutes a conventional precision metric. A set of ontologies with different quality aspects and of different granularity was selected (see Appendix I).

Phase III: Prototype II

The objective of this phase was to get a better understanding of the sensitivity of the FVC components with respect to ontology quality. Therefore, a new prototype was implemented and a set of experiments was conducted in a controlled environment (Wohlin et al., 2003) based on the lessons learned from the user experiment conducted in Phase II. Furthermore, to evaluate the quality of the FVs, a set of FV quality measures was proposed and validated.

The research methods used in this phase were engineering and experimentation, but also influenced by the results from the previous phases. Engineering was used to construct an alternative FVC approach to the one created in Phase II. The new approach was based on lessons learned. A set of experiments was conducted and the results analysed with respect to both FV quality and ontology quality. The quality was measured using

intrinsic and extrinsic measures with respect to ontologies of different granularity. The proposed measures were validated in an experiment with real users. The same ontologies as in Phase II were used.

3 Related Work

In this overview, we discuss search approaches using Semantic Web (SW) techniques (e.g. ontologies) in general, though emphasis is placed on approaches that construct FVs and their use in search. This synopsis provides a more comprehensive overview of related work than that found in the papers presented in Part II. First, we introduce the Semantic Web and categorise approaches to semantic searches. Then, we explore information retrieval approaches that are using semantic techniques to improve retrieval effectiveness followed by approaches to FVC and similar. Before highlighting key points at the end of this chapter, we provide a brief overview of approaches for the evaluation of semantic search systems.

3.1 Introduction

The Web contains vast resources of information, and the diversity of topics and terminologies makes it difficult to find relevant information on the Web (Ding et al., 2005; Horrocks, 2007). Recall Section 1.1, where we presented word ambiguity as one of the core problems in finding relevant information. The Semantic Web (Berners-Lee et al., 2001) is believed to extend the current Web and provide a means to tackle some of these difficulties (Horrocks, 2007; van Harmelen, 2006). The grand idea is to annotate every piece of information with machine-processable semantic descriptions to enable a more advanced usage of information elements like reasoning.

There is a diversity of definitions for semantic search. For instance, Guha et al. (2003) define semantic search as "an application of the Semantic Web to search". This is limiting semantic search to the Semantic Web, and hence does not represent the diversity of semantic search systems found on the Web. Wang et al. (2008), on the other hand, states that "semantic search supplements and improves conventional information retrieval systems on the basis of structural knowledge representation formalisms". We adopt this definition in this work since it better fits the diversity of semantic search systems available on the Web. In any case, a core functionality of semantic search engines is word disambiguation. Furthermore, many commercial semantic search systems usually merge information from several external sources into one unified view of the retrieved information (see Section 3.2.2).

Search is one of several applications for the Semantic Web. There are many approaches to semantic search, e.g. some rely on semantic annotations (Yang, 2006) while others enhance clustering of retrieved documents (Panagis et al., 2006). In (Strasunskas & Tomassen, 2010), we classified semantic search based on an analysis of reviewed literature and related classification schemes (summarised in Figure 3.1). As can be seen from the figure, search applications can be categorised along seven dimensions. However, to be classified as a semantic search application, w.r.t. the definition previously presented, the system must utilise some form of structural knowledge that is used to improve the retrieval effectiveness (i.e. relevance and/or user experience). In the following, we elaborate on each of the different aspects of semantic search systems.

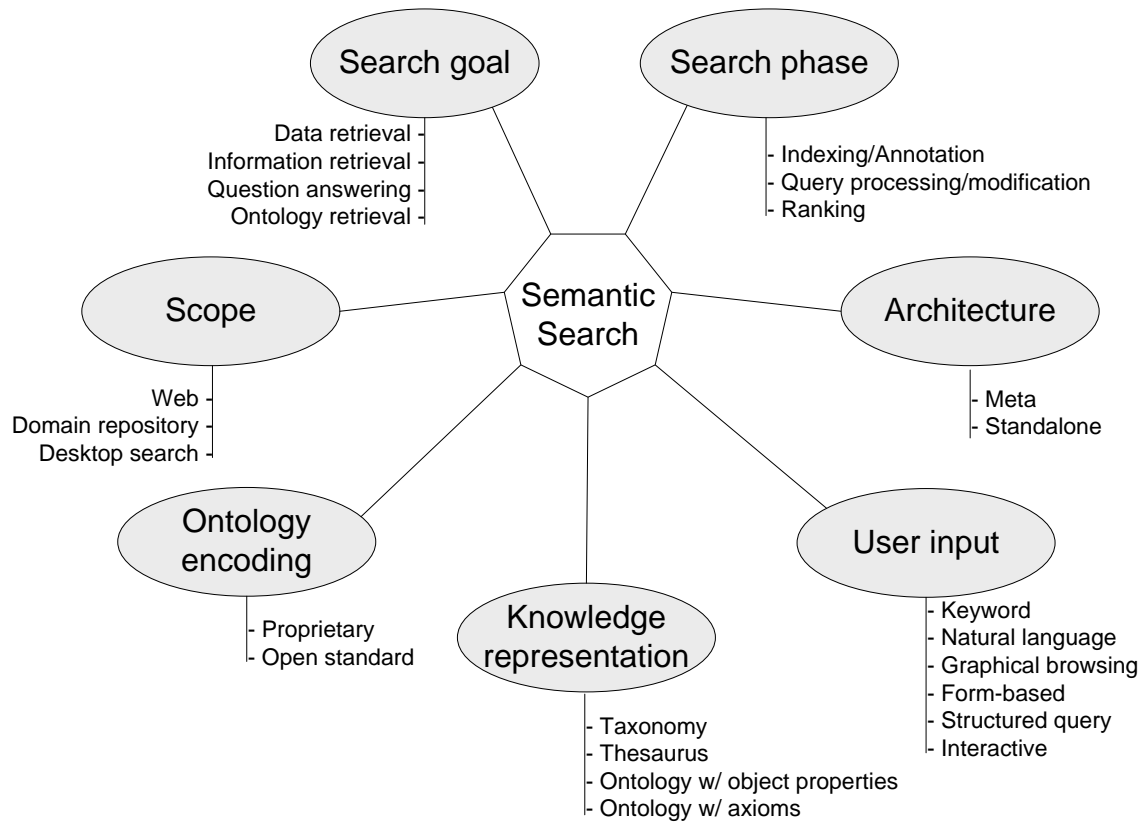


Figure 3.1: Aspects of semantic search systems (adopted from (Strasunskas & Tomassen, 2010)).

Search phase

Most semantic search applications are based on semantic annotation. Typically, documents as whole, or document elements (e.g. named entities), are annotated with meta-data. In any case, the documents or elements are normally treated as ontology instances (Castells et al., 2007; Kiryakov et al., 2004; Rocha et al., 2004; Song et al., 2005). Many other approaches focus on query processing and query expansion (Bhogal et al., 2007; Chang et al., 2006; Ciorascu et al., 2003; Grootjen & van der Weide, 2006; Rajapakse & Denham, 2006). The aim is to disambiguate the users' queries by adding domain specific terms, synonyms, etc. Furthermore, there are approaches focusing on filtering and ranking of retrieved documents (Anyanwu et al., 2005; Braga et al., 2000; Ding et al., 2005; Stojanovic et al., 2003).

Architecture

Semantic search systems are in general either standalone or meta-search engines. Standalone systems are typically implemented for a specific domain or intranet/desktop search (Chirita et al., 2006; Zhang et al., 2005) since there is limited annotated information available on the Web. While meta-search engines function on top of existing Web search engines, and mainly extend existing systems with semantic query expansion or semantics-based document re-ranking (Burton-Jones et al., 2003; Stojanovic et al., 2003). There are also hybrid systems that try to combine the best of both worlds (Amaral et al., 2004; Harth et al., 2007).

User input

Semantic search systems can be categorised according to the complexity of their required user interaction as follows:

- **Keyword based queries.** The user can enter keywords in a simple text field. This is probably the most common form of user query entry for Web search engines today. For semantic search engines, the queries are typically enriched using background knowledge, i.e. ontologies (Bhogal et al., 2007; Castells et al., 2007; Ciorascu et al., 2003).
- **Natural language based queries.** Keyword based queries are heavily used but still constitute an artificial way of expressing information needs, while form based or structured queries tend to have a complex syntax. Therefore, there are approaches focusing on enabling natural language interfaces to specify queries and obtain answers (Lopez et al., 2007; Tablan et al., 2008).
- **Graphical browsing.** Graphical ontology browsing can be an intuitive interface for novice end-users to ontologies, but may often require more interaction by the users (Brasethvik, 2004; Suomela & Kekalainen, 2005).
- **Form based queries.** Form based queries typically include the possibility to create more specific or restrictive queries compared to keyword based queries. Often the user can restrict the query to one or more specific field, since these approaches are tailored for a specific domain (Aitken & Reid, 2000; Kim, 2005; Ungrangsi et al., 2007).
- **Structured queries.** Formal structure query specification targets, by default, experienced users or software agents (Blacoe et al., 2008; Wang et al., 2008). Typically a knowledge based approach to interact with the information is adopted (i.e. using reasoning mechanisms and ontological query languages like SPARQL, RDQL, OWL-QL) to retrieve instances (Blacoe et al., 2008).
- **Interactive queries.** The idea of interactive queries is to involve the users in the search process (e.g. "I want to have tables", "What colour?", "Red") in an attempt to improve the final search results (also referred to as relevance feedback (Manning et al., 2008, p. 163)). Traditionally, an external source (e.g. a thesaurus or WordNet (Fellbaum, 1998)) is included in the reformulation process while semantic search systems typically use knowledge models, i.e. ontologies (Bhogal et al., 2007; Burton-Jones et al., 2003; Nagypal, 2005; Suomela & Kekalainen, 2005).

Knowledge representation

Common to all semantic search systems is to include one or more knowledge models. Early work focused on taxonomy and thesaurus usage in order to improve searches (Aitken & Reid, 2000; Ciorascu et al., 2003), while recent developments have employed richer knowledge structures using object properties, axioms and instances (Lopez et al., 2007).

Ontology encoding

There is a variety of different ontology languages; some are proprietary encoding formats but most are open standards like OWL (McGuinness & van Harmelen, 2004) or RDFS (Brickley & Guha, 2004). In addition, there are other formats, typically a result

of academic developments (often open formats although not considered standard) like DAML-OIL (Connolly et al., 2001). For examples of usage, proprietary ontology languages are used in (Amaral et al., 2004), open standards are used in (Chirita et al., 2006; Guha et al., 2003), while other formats are used in (Brasethvik, 2004; Burton-Jones et al., 2003; Zhang et al., 2005).

Scope

Semantic search systems target the same scopes as traditional search systems like the Web, domain repositories and desktop searches. Web search is tackled by (Corby et al., 2006; Rocha et al., 2004). Both desktop search and domain repositories are addressed by (Kiryakov et al., 2004), while (Castells et al., 2007) improve domain repositories and (Chirita et al., 2006) specifically focus on desktop search.

Search goal

A search system is designed to satisfy the information needs of its end-users. However, most systems are tailored to perform specific tasks since it is difficult, or impossible, to design a system that satisfies all information needs. Therefore, systems are in general designed to either perform data retrieval (Guha et al., 2003; Ning et al., 2009), information retrieval (Formica et al., 2008; Paralic & Kostial, 2003), question answering (Frank et al., 2007; Lopez et al., 2007), or ontology retrieval (Pan et al., 2006; Ungrangsi et al., 2007).

Furthermore, the search process can be decomposed into specific search tasks based on the information needs of the end-users. For instance, Guha et al. (2003) distinguish two different kinds of search, namely navigational search (i.e. the user is using a search engine to navigate to a particular document providing required information) and research search (i.e. the user is trying to locate a collection of documents). While Aula (2003) classified search tasks into three categories: fact-finding, exploratory and comprehensive search tasks. In fact-finding, a precise set of results is more important than the amount of retrieved documents. In exploratory search tasks, the user wants to obtain a general understanding about the search topic; consequently, neither high precision nor recall is more important. Finally, in comprehensive search tasks, the concern is to find as many documents as possible on a given topic; therefore, both recall and precision should be as high as possible. Typically, search systems for the SW tend to focus on fact-finding (Kiryakov et al., 2004; Schumacher et al., 2008).

3.2 Semantics in Information Retrieval

There are many approaches to semantic searches using ontologies to improve the retrieval effectiveness of searches, such as treating documents as instances or annotating them using ontology instances (Castells et al., 2007; Kiryakov et al., 2004). Castells et al. (2007) use weighted annotation when associating documents with ontology instances. The weights are based on the frequency of occurrence of the instances in each document. They report measurable improvements with their approach compared to traditional keyword-based searches. The reader is referred to (Reeve & Han, 2005; Uren et al., 2006) for reviews of similar approaches. Others use ontologies for the representation of concepts (Ozcan & Aslangogan, 2004). Ozcan and Aslangogan

(2004) extend each concept with similar words using a combination of Latent Semantic Analysis (LSA) and WordNet (Fellbaum, 1998). Testing shows promising results for short or poorly formulated queries. While some approaches focus on using ontologies in the process of enriching queries (Ciorascu et al., 2003; Paralic & Kostial, 2003). However, ontologies in such cases typically serve as thesauri containing synonyms, hypernyms/hyponyms, and do not consider the context of each term (i.e. every term is equally weighted).

In the following subsections, we provide an overview of semantic-based information retrieval systems that utilise semantic techniques to enhance searches. The overview is limited to approaches that endeavour to make improvements by employing analysis of semantics rather than by taking different measures or inherent semantic from texts (e.g. Latent Semantic Analysis/ Indexing (Manning et al., 2008, p. 378), Meaning-Text Theory (Melchuk, 1981)). By different measures we mean analysis of Web content with respect to information quality, often used for ranking purposes in order to improve precision (e.g. approaches such as PageRank (Page et al., 1999) based on references among Web pages, ranking based on information updates). More specifically, we focus on systems meeting one or more of the following criteria with respect to the categorisation summarised in Figure 3.1:

- **Architecture:** Meta, standalone
- **User input:** Keyword
- **Knowledge representation:** Ontology
- **Scope:** Web, domain repository
- **Search goal:** Information retrieval

In Section 3.2.1, we provide an overview of academic approaches to ontology-based information retrieval found relevant to this work, while in Section 3.2.2 we provide a brief overview of some commercial semantic search systems. There are approaches not using ontologies but still related to this work, especially approaches for query refinement (a brief overview of such related approaches is provided in Section 3.2.3). Finally, in Section 3.2.4 we explore related work on the construction of feature vectors.

The reader is also referred to (Esmaili & Abolhassani, 2006; Mangold, 2007; Scheir et al., 2007; Strasunskas & Tomassen, 2010) for other overviews of semantic search systems.

3.2.1 Academic approaches

In this section, we explore related work on the enhancement of searches where ontologies are used. Typically, information retrieval systems make use of ontologies to help the users clarify their information needs and come up with semantic representations of documents. The basic assumption of ontology-based information retrieval (ObIR) systems is as follows:

If a person is interested in information about B, it is likely that she will find information about A interesting, provided that A and B are closely related concepts in an ontology (i.e. these systems exploit semantic relationships).

In the simplest way, a user's query is expanded by hypernyms/super-classes, i.e. generalisation (Bonino et al., 2004), or hyponyms/sub-classes, i.e. focalisation (more detailed knowledge) (Bonino et al., 2004), or other related concepts (e.g. sibling concept and other neighbourhood concepts). Below, we discuss some important ontology-based information retrieval approaches relevant to this work.

OntoSearch by Jiang and Tan (2006) is a full text search engine that depends on documents annotated with elements from an ontology (i.e. if a concept is specified in a document then it is associated with it). The user submits a traditional keyword-based query that yields an initial set of documents (see Figure 3.2). These retrieved documents contain semantic annotations (i.e. concepts) that are used to obtain a set of associated concepts. The spreading activation algorithm uses this set of associated concepts to infer those concepts that are semantically related to the initial set of concepts (i.e. from the retrieved documents). Consequently, the most relevant concepts are determined through the inference process of the algorithm. The conceptual relevance scores obtained by the spreading activation algorithm are used to re-rank the retrieved documents before presenting them to the user. The classical cosine measure is used to calculate the similarity between the documents and queries. Results show that the approach performs better than a comparable keyword-based approach.

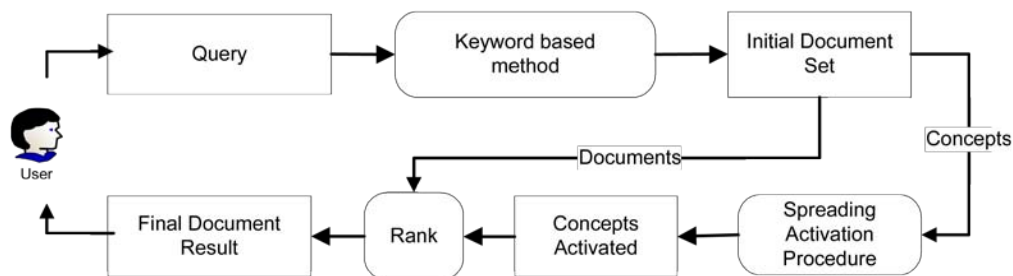


Figure 3.2: The system flow of OntoSearch (Jiang & Tan, 2006).

PowerMap by Lopez et al. (2006b) is an ontology mapping application. PowerMap is the core component of the PowerAqua question answering system by Lopez et al. (2006a). PowerAqua follows an earlier system called AquaLog (Lopez et al., 2007) and addresses some of its shortcomings. PowerAqua takes natural language queries as input, while the input query to PowerMap must be formulated as a triple. PowerMap functions as follows, first the query is analysed and expanded with corresponding synonyms, hypernyms and hyponyms from WordNet (Fellbaum, 1998) in combination with SUMO (an upper level ontology proposed by (Niles & Pease, 2001)) to bridge the gap between the user terminology and the terminology of the ontologies. This extended query is used to retrieve a set of candidate entities. Next, they perform semantic mapping of the entities to filter out those of low relevance and clustering of the senses to disambiguate the query terms. Finally, the relations of the candidates are considered and the most prominent ontologies are presented. PowerMap can deal with several heterogeneous ontologies that can be discovered based on the content of the user's query. The most prominent ontologies are used by PowerAqua (Lopez et al., 2006a) to extract answers relevant to the user's query expressed in natural language.

Braga et al. (2000) are using ontologies for retrieval and filtering of domain information within or across multiple domains. Each ontology concept is defined as a domain

feature with detailed description relevant to the domain including relationships with other features. The Feature Model deals with relationships between domain features. However, the ontology usage is limited to hypernyms (super class), hyponyms (sub class), and synonyms.

Formica et al. (2008) propose a novel way of ranking annotated documents with respect to both an ontology and a user query. The documents have been annotated with a set of characterising concepts in advance, called feature vectors, which are assumed to be already built. Figure 3.3 visualises the relationship between a concept and its extensions. They distinguish between two types of extension, Feature Extension (FE) and Similarity Featured Extension (SFE). FE represents all the recourses in a Universe of Digital Resources (UDR) containing concept c , while SFE represents all the recourses in UDR with a similarity with respect to concept c above a certain threshold. Consequently, these feature vectors function as instances of the corresponding concepts. Next, they calculate the similarity between the concepts of a user query and the feature vectors with respect to an ontology. Testing shows that their approach performs slightly better than other comparable approaches. However, calculation of the similarity scores is limited to the hierarchical structure of the ontology.

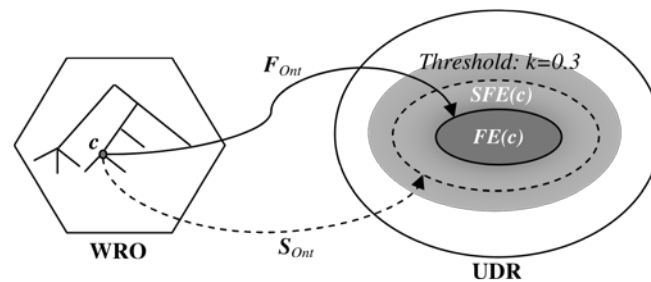


Figure 3.3: The relationship between concepts and extensions (Formica et al., 2008).

Nagypal (2005) proposes a general framework based on ontology-supported semantic metadata generation and ontology-based query expansion. One of the strengths of the framework is that it is capable of handling imperfect ontologies. The framework targets the transition from the current Web to the Semantic Web by extending existing search engines. Nagypal proposes to instantiate the ontology with the terms found in the search engine index. The ontology is used during the query formulation to disambiguate queries. In the case of an ambiguous term, a list of selectable alternative interpretations is shown to the user. Various ontology-based heuristics are applied to the query creating a set of queries. These queries are submitted to the underlying search engine and the results are combined by the use of Bayesian network techniques. The search process is depicted in Figure 3.4. Nagypal (2007) also found that ontology quality has a significant effect on the retrieval performance.

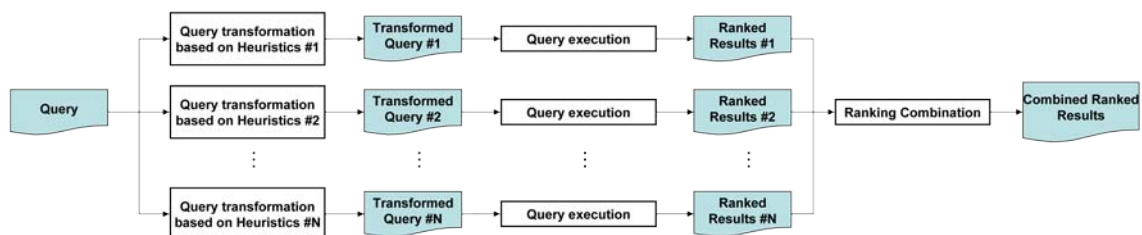


Figure 3.4: Illustration of the search process (Nagypal, 2005).

Paralic and Kostial (2003) propose an ontology-based approach to information retrieval where resources (i.e. documents) are associated with concepts in an ontology. The focus of their research is query processing, which is depicted in Figure 3.5. The concepts in a query are matched to corresponding concepts in an existing ontology. Then the query concepts are matched with the document concepts and matched documents are retrieved. Finally, the total similarity score is calculated. When compared to the vector model, TF-IDF and the Latent Semantic Indexing (LSI) approach, their ontology-based approach performed significantly better.

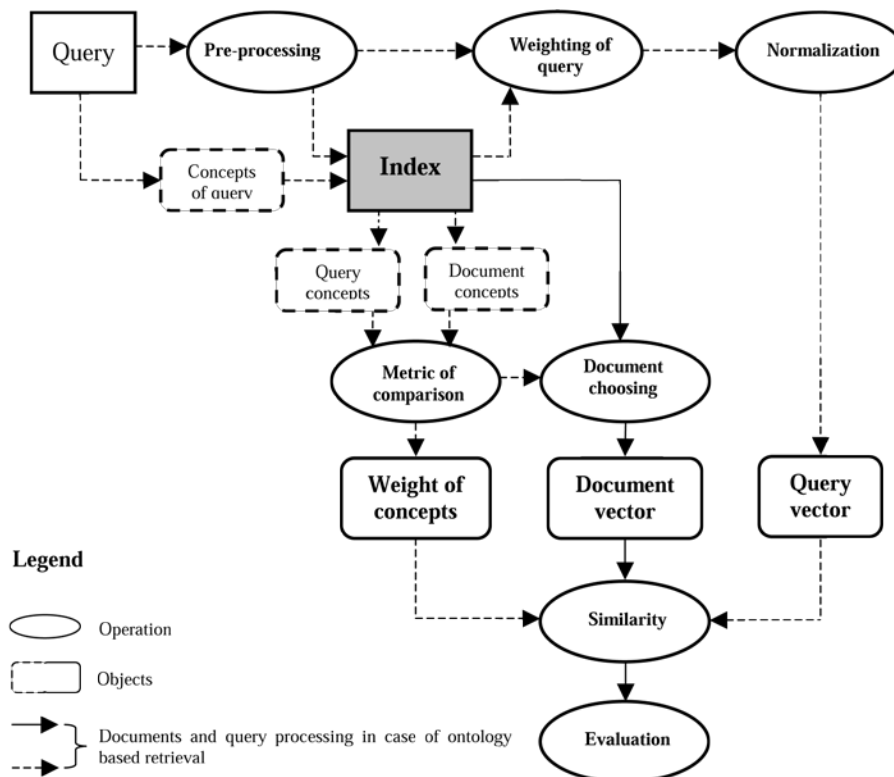


Figure 3.5: The proposed approach to query processing (Paralic & Kostial, 2003).

Zhou et al. (2007) propose a Topic Signature Language Model that is used to perform semantic smoothing to increase retrieval performance. They create topic signatures for each concept defined in a domain specific ontology using a highly relevant document collection. The topic signature terms are found by collocation. They assume the concepts are unique and consequently circumvent the problem of word disambiguation. For general domains where no ontology exists, they propose to use multiword expressions as topic signatures. The multiword expressions contain context and are consequently less ambiguous. They report significant improvement over comparable language models.

Harth et al. (2007) propose an entity-centric search engine called Semantic Web Search Engine (SWSE). They semantically integrate structured data from both static and live sources into a coherent knowledge base. The knowledge is stored as a large graph of RDF entities and hence provides an entity-centric approach versus the more traditional document-centric approach. The information needs are formulated by keywords. The user can filter the search results by entity type and navigate between entities. Figure 3.6

depicts the high-level architecture with the data flow within the system. The *Semantic Search and Query Engine* is the core component of the system. It contains the RDF data store and provides the entity search and navigation interface to the knowledge base. The *Data Preparation and Integration* provides adapters to a multitude of different formats. The crawler extracts metadata and converts it to RDF where necessary. Finally, the *On-Demand Integration* component provides wrappers for querying external sources. A demo of the search engine is available¹.

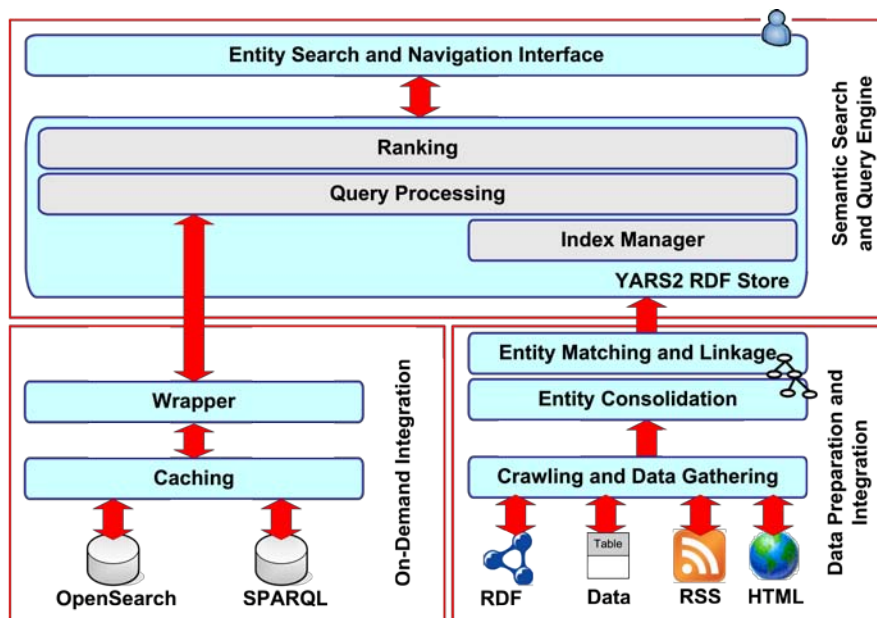


Figure 3.6: The architecture of the Semantic Web Search Engine (Harth et al., 2007).

The various semantic search systems presented in this section are summarised in Table 3.1. As can be seen from the table, most of the reviewed approaches utilise ontologies to enhance search results. Many use the ontologies to annotate documents in the indexing process and consequently most of the systems are either standalone or hybrid. Furthermore, many do not focus on the user interface and hence provide only keyword user input. Since one or more knowledge models are used they all target one or more domains. Some also target the Web, to show the applicability of their approach.

Table 3.1: Summary of reviewed academic approaches to semantic searches.

	<i>Search phase</i>	<i>Arch.</i>	<i>User input</i>	<i>Knwl. rep.</i>	<i>Ont. enc.</i>	<i>Scope</i>	<i>Search goal</i>
Braga et al. (2000)	I, R	Hybrid	Graphical	Ontologies	Proprietary	Web, Domain	IR
Formica et al. (2008)	R			Taxonomy		Domain	DR
Harth et al. (2007)	I, R	Hybrid	Keyword	Ontologies	Open std. (RDF)	Web, Domain	DR

¹ Semantic Web Search Engine (Available from: <http://swse.deri.org/>).

	<i>Search phase</i>	<i>Arch.</i>	<i>User input</i>	<i>Knwl. rep.</i>	<i>Ont. enc.</i>	<i>Scope</i>	<i>Search goal</i>
Jiang and Tan (2006)	I, Q, R	Stand-alone	Keyword	Ontologies	Proprietary	Domain	IR
Lopez et al. (2006a; 2006b; 2007)	I, R	Stand-alone	Natural language	Ontologies	Proprietary (OCML) Open std. (RDF, OWL)	Web, Domain	QA
Nagypal (2005)	I, Q, R	Stand-alone	Interactive	Ontologies	Open std. (OWL)	Domain (Wikipedia)	IR
Paralic and Kostial (2003)	I, R	Stand-alone	Keyword	Ontologies	Proprietary	Domain	IR
Zhou et al. (2007)	I, R	Stand-alone	Keyword	Ontologies	Proprietary	Web (?), Domain	IR

3.2.2 Commercial approaches

Here we provide a brief overview of commercial semantic search engines currently available on the Web and this section is meant to provide an insight into the diversity of the systems. A broader list of semantic search systems in general is found on HLWIKI², while SWUI-wiki³ provides an overview of academic approaches to semantic searches. Following, short descriptions of some of the commercial semantic search systems are provided, before we summarise the findings with respect to the categorisation scheme presented in Section 3.1.

Hakia

<i>Provider</i>	<i>URL</i>
Hakia Inc.	http://www.hakia.com

hakia.com is a general purpose semantic search engine provided by Hakia Inc. Hakia segments the search results into different categories, including News, Twitter, Blogs, Images, Video, Wiki, Galleries, Web, Credible, and Pubmed. Credible include results from trusted sources that have been approved by Hakia (or on behalf of), while Pubmed includes results from the MEDLINE⁴ database. For ambiguous queries, Hakia provides categorised results, called Galleries. These categories are created offline and semi-automatically to ensure high quality. For each of the results presented in Galleries a resume is provided. In addition, Hakia includes an excerpt from Wikipedia and results from Twitter when appropriate. The user can only formulate queries as simple keywords.

A screenshot of the Hakia semantic search engine is shown in Figure 3.7. Each category can be expanded or collapsed at will, while categories with no results are hidden from the user. For each search result a small contextual text fragment is shown.

² HLWIKI (Available from: http://hlwiki.slais.ubc.ca/index.php/Semantic_search).

³ SWUI wiki (Available from: http://swuiwiki.webscience.org/index.php/Semantic_Search_Survey).

⁴ MEDLINE is a bibliographic database covering health related information compiled by the United States National Library of Medicine.

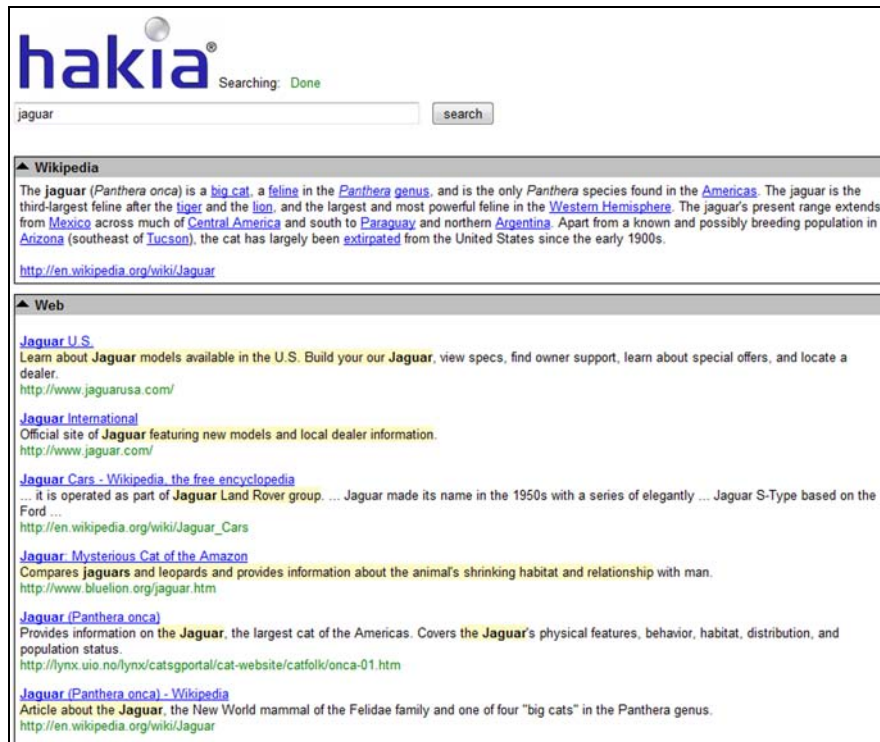


Figure 3.7: A screenshot of Hakia.

Powerset

Provider	URL
Microsoft	http://www.powerset.com

Powerset is a semantic search engine provided by Microsoft. The search results from Powerset come from Wikipedia. "Factz" represents the core elements of Powerset. A Factz is a triple consisting of a *thing* that is *related* to another *thing*. They are extracted from Wikipedia articles. Note that one Factz does not necessarily represent the truth, but combined with related or equal Factz they can aggregate results of high probability. Powerset also incorporate results from Freebase⁵ that are used to provide instant answers and dossiers. In addition, the user currently has the option of submitting the same query to Bing (i.e. a search engine for the Web provided by Microsoft). The user can formulate queries as simple keywords, phrases, or questions. Some simple questions can be answered directly, like "What is the capital of Norway?" with the correct answer "Oslo" while "What is the capital city of Norway?" provided a list of results.

A screenshot of Powerset is shown in Figure 3.8. As can be seen, results from Freebase are presented at the top, while Factz are presented just below the results from Freebase. Below the results from Freebase and the Factz are the Wikipedia articles listed. Ambiguity is handled by grouping the various senses into tabs.

⁵ Freebase is an open repository of structured data (Available from: <http://www.freebase.com>).

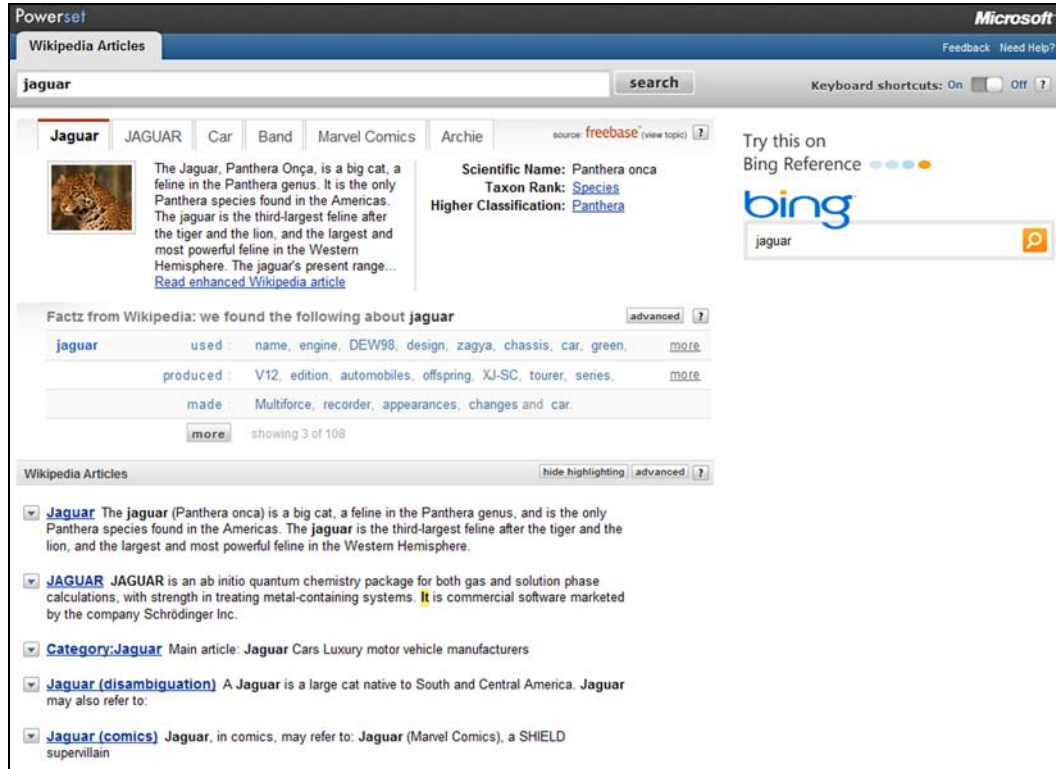


Figure 3.8: A screenshot of Powerset.

SenseBot

<i>Provider</i>	<i>URL</i>
Semantic Engines LLC	http://www.sensebot.net

SenseBot is a meta-search engine by Semantic Engines LLC. It currently uses Google, Yahoo! and Bing as backend search engines. The users also have the option to use SenseBot, but whether it has an own index is unknown. Instead of delivering a ranked list of links with a small text summary of each of the documents, it provides a summary on the topic of the users query. The summary is presented as a tag cloud depicting the most significant concepts of the topic. The tag cloud is generated based on the content of the top ranked documents using text-mining techniques to analyse the Web pages and hence identify the foremost concepts. The concepts in the tag cloud can be used to refine the searches. In addition, a list of selected sentences in conjunction with their source is presented. The sentences are those believed to be most relevant with respect to the topic of the query. The queries can only be formulated using simple keywords.

Figure 3.9 depicts a screenshot of SenseBot and the search results of our standard ambiguous query, "jaguar". As seen in the figure, the main search result is the tag cloud performing as a summary of the search result. For ambiguous queries like "jaguar", the summary can contain a mix of concepts from many different domains as seen in Figure 3.9. Selecting a concept in the tag cloud expands the current query and hence refines the search.

SenseBot
The Search Engine that finds sense in a heap of Web pages

Save summary Modify summary 20 sentences Show images Help

[ANIMALS](#) [ARIZONA](#) [CALLING](#) [CARFINDER](#) [CAT](#) [COMMITMENT](#)
[DEALERSHIP](#) [DIRECTIONS](#) [DRIVING INSTRUCTIONS](#) [EAGER](#) [ENTHUSIASM](#)
[EXCEEDING](#) [EXPERIENCED SALES STAFF](#) [FINANCING](#) [HIGH EXPECTATIONS](#)
[INDIVIDUAL CUSTOMER](#) **[JAGUAR](#)** [MATCHING CAR](#) [MEETING](#) [MEXICO](#)
[NOTIFICATIONS](#) [ONLINE FORM](#) [ONLINE INVENTORY](#) [PANTHERA ONCA](#)
[PARAMOUNT](#) [PREY](#) [REQUEST](#) [SCHEDULE](#) [SERVING](#) [SPECIES](#)
[STANDARDS](#) [UNITED STATES](#)

SUMMARY: "jaguar"

Use the CarFinder to select the [Jaguar](#) of your choice and to receive notifications on when it has arrived.
[SOURCE: [Jaguar Tacoma | New Jaguar dealership in Fife, WA 98424 \(www.jaguartacoma.net/index.htm\)](#)]

Once found in wooded regions from the U.S.-Mexican border south to Patagonia, the [jaguar](#) (*Panthera onca*) survives, in reduced numbers, only in remote areas of Central and South America; the largest known population is in the Amazon rain forest.
[SOURCE: [Jaguar: Definition from Answers.com \(www.answers.com/topic/jaguar\)](#)]

To visit us and test drive a [Jaguar](#), click on [Dealership](#): Directions for step-by-step driving instructions to our [dealership](#), or give us a call.
[SOURCE: [Jaguar Tacoma | New Jaguar dealership in Fife, WA 98424 \(www.jaguartacoma.net/index.htm\)](#)]

Apart from a known and possibly breeding population in Arizona (southeast of Tucson), the [cat](#) has largely been extirpated from the United States since the early 1900s.
[SOURCE: [Jaguar \(Panthera onca\) - Wikipedia \(en.wikipedia.org/wiki/Jaguar\)](#)]

Javelina and deer are presumably mainstays in the diet of jaguars in the United States and Mexico borderlands. [...] In Arizona, jaguars ranged widely throughout a variety of habitats from Sonoran desert scrub upward through subalpine conifer forest.
[SOURCE: [Jaguar: Definition from Answers.com \(www.answers.com/topic/jaguar\)](#)]

The [jaguar](#) (*Panthera onca*) is a big [cat](#), a feline in the *Panthera* genus, and is the only *Panthera* species found in the Americas.
[SOURCE: [Jaguar \(Panthera onca\) - Wikipedia \(en.wikipedia.org/wiki/Jaguar\)](#)]

Figure 3.9: A screenshot of SenseBot.

True Knowledge

Provider	URL
True Knowledge Ltd.	http://www.trueknowledge.com

True Knowledge (TK) is a new semantic search engine by True Knowledge Ltd. (it is currently in beta). TK is a question answering (QA) application targeting the Web. True Knowledge has many similarities with Freebase⁵, they both scrape the Web for facts (e.g. from Wikipedia) and the community can add additional facts. The facts are probably stored in a similar fashion to Powerset - that is, using triplets (i.e. TK provides a *relation finder* service where users can search for different relations used by TK, e.g. "is married to"). In addition, TK also provides traditional search results. Users can formulate their queries in natural language but keywords are also supported. Furthermore, when TK provides a direct answer to a question (e.g. to "Who was the president of the US in 1851?" TK answers "Millard Fillmore"), it also provides a reason for that conclusion (e.g. "Millard Fillmore was the president of the United States between July 9th 1850 and March 4th 1853").

Either the answer to a query can be presented as a list of facts (see Figure 3.10) or as a more precise answer (e.g. for the query "Is a jaguar a human being?" TK answers "No"). Furthermore, in some cases TK also presents a traditional list of relevant search results (as was the case for the "jaguar" query seen in Figure 3.10, but not shown in this figure).

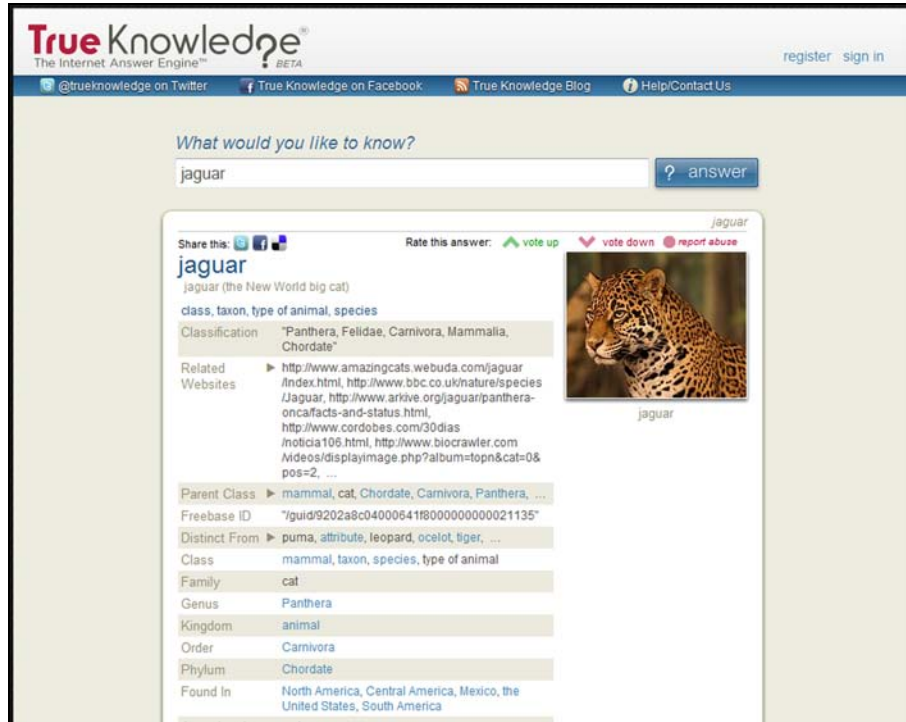


Figure 3.10: A screenshot of True Knowledge.

Yebol

<i>Provider</i>	<i>URL</i>
Yebol	http://www.yebol.com

Yebol is a categorisation based semantic search engine. It is inspired by the Yahoo! Directory, which is a manually constructed directory of the Web. However, the amount of information on the Web has grown to a proportion that no longer makes this approach feasible. Therefore, Yebol plan to automate the building of a similar knowledge base by combining human labelled information with a set of sophisticated algorithms (i.e. of association, clustering, and categorisation). The goal is to automatically generate knowledge for search concepts, Web sites, Web pages and users. To accomplish this, they are using Amazon.com cloud computing services to build a large knowledge base (currently there are more than 10 million concepts and 1 billion web pages).

Yebol is similar to Hakia in the sense that they strive to categorise the Web. Yebol currently has the following categories: News, Twitter, Videos, Images, Top Sites, Categories, Search Results, and Related Searches. In addition, they provide traditional search results for queries that cannot be answered by their knowledge base. Most search engines present their search results in a linear structure, while Yebol presents their categories in a tree like structure (see Figure 3.11), creating a dense overview of relevant results.

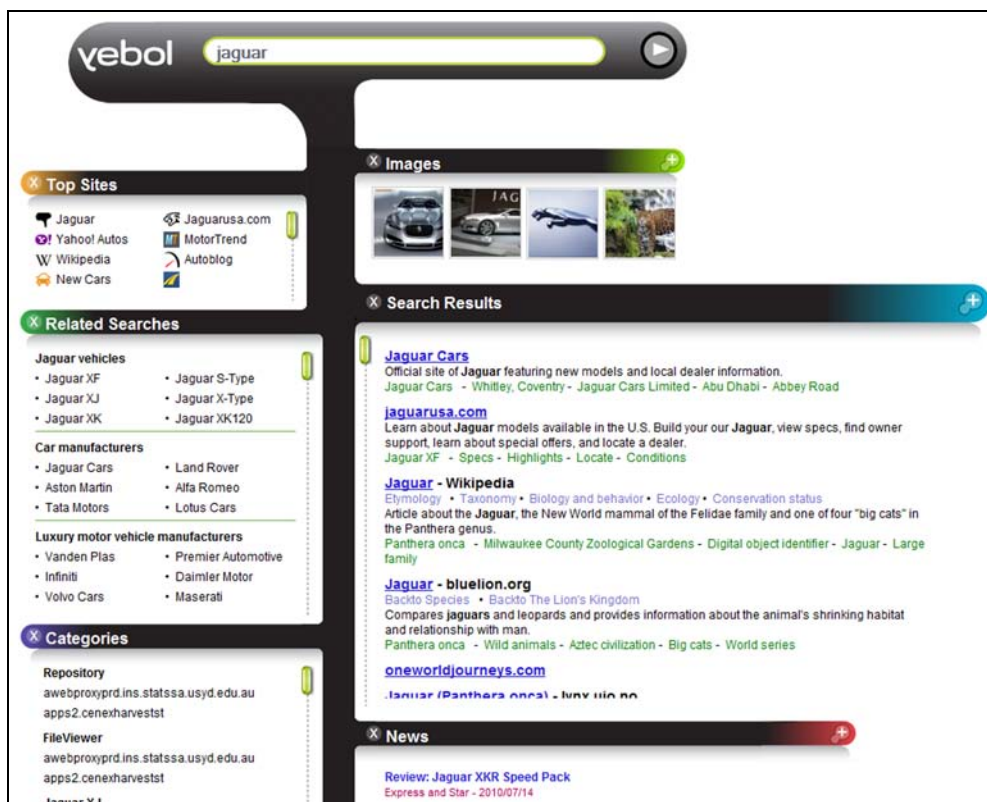


Figure 3.11: A screenshot of Yebol.

The various semantic search engines presented in this section are summarised in Table 3.2. Since the search engines reviewed are commercial, there is limited information available on the business critical features of their solutions. All reviewed systems, except for SenseBot, were hybrid systems and consequently alter all the search phases (it is unclear whether SenseBot has its own index). They typically merged information from many various sources into one combined view. All systems accept keyword-based user queries, while two systems (Powerset and True Knowledge) also target natural language queries (True Knowledge being most advanced). Two of the systems targets query answering and their underlying knowledge is represented as triplets, though for True Knowledge this was not explicitly stated. Most of the search engines also target the Web, with the exception of Powerset which only targets a specific domain (i.e. Wikipedia).

Table 3.2: Summary of reviewed commercial approaches to semantic search.

	<i>Search phase</i>	<i>Arch.</i>	<i>User input</i>	<i>Knwl. rep.</i>	<i>Ont. enc.</i>	<i>Scope</i>	<i>Search goal</i>
Hakia	I, Q, R	Hybrid (Twitter)	Keyword			Web, Domain (Wikipedia, PubMed)	IR
Powerset	I, Q, R	Hybrid (Freebase)	Keyword, Natural language	Other (Factz)		Domain (Wikipedia)	IR, QA
SenseBot	Q, R	Meta (Google, Yahoo!, Bing)	Keyword			Web	IR

	<i>Search phase</i>	<i>Arch.</i>	<i>User input</i>	<i>Knwl. rep.</i>	<i>Ont. enc.</i>	<i>Scope</i>	<i>Search goal</i>
True Knowledge	I, Q, R	Hybrid (Unknown)	Keyword, Natural language	Other (Triples)		Web	IR, QA
Yebol	I, Q, R	Hybrid (Yahoo!, Twitter)	Keyword			Web, Domain (Twitter)	IR

3.2.3 Query reformulation approaches

In this section, we present an overview of approaches to query reformulation that do not necessarily use ontologies but still are relevant to this work. There are many approaches to query reformulation, especially query expansion (Adi et al., 1999; Carpineto & Romano, 2010; Chang et al., 2006; Grootjen & van der Weide, 2006; Qiu & Frei, 1993; Rajapakse & Denham, 2006). The aim of query expansion is to enhance the initial query by adding new and meaningful terms (Bhogal et al., 2007). This is typically done by extending the query terms provided with synonyms or hyponyms (Chenggang et al., 2001). Next, we provide a small overview of query reformulation approaches related to this work. The reader is referred to (Bhogal et al., 2007) for a review of ontology based query expansion approaches.

Adi et al. (1999) present a commercial search engine that provides three basic search strategies: word, concept, and super-concept. A concept is represented as a set of words, while a super-concept is a combination of several closely related concepts. The user may mix strategies when searching. Unfortunately, there are not enough details available in (Adi et al., 1999) to state how this works.

The work by Chang et al. (2006) rely on query concepts. Two techniques are explored to create the feature vectors of the query concepts, based on a document set (i.e. globally) and result set of a user query (i.e. locally). Experimental evaluation shows that the approach is as good as current query reformulation approaches, and especially effective for short or poorly formulated queries. Furthermore, they found the performance of the approach most effective when concepts were generated from retrieved documents instead of a document collection, which backed their previous findings reported in (Chang et al., 2004).

Qiu and Frei (1993) are using query expansion that is based on a similarity thesaurus. The similarity thesaurus reflects the domain knowledge and is automatically created. Weighting of terms is used to reflect the domain knowledge. Query expansion is based on similarity between terms, in document space and query concepts (depicted in Figure 3.12, where query concept q_c is most similar to term t_1) in contrast to all query terms. They report an improvement of around 20-30% when compared to simple term based queries, especially for shorter queries.

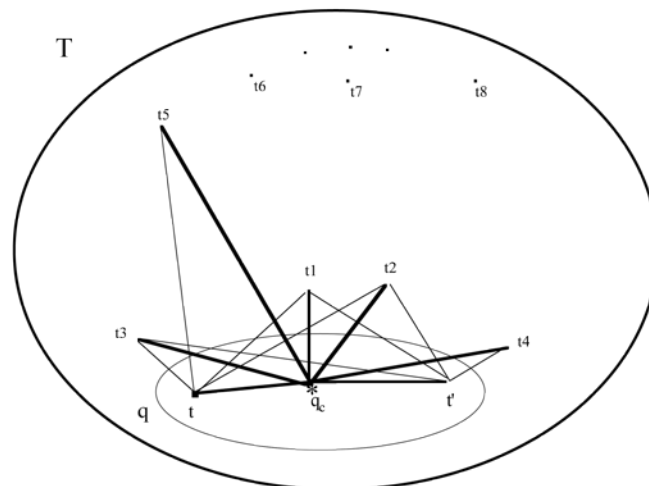


Figure 3.12: Relationships between terms and query in document vector space (Qiu & Frei, 1993).

Grootjen and van der Weide (2006) propose a similar approach to Qiu and Frei (1993). They describe a conceptual query expansion approach where query concepts are based on search results. The query concepts are created from an initial result set that is projected onto global information (e.g. a thesaurus) yielding a local conceptual view. The selection of candidate concepts for query refinement is based on this local view. Furthermore, only those query terms that in combination make sense in the document collection are considered for query expansion. The approach reports an improvement especially for short or poorly formulated queries.

Qiu and Frei (1993) and Grootjen and van der Weide (2006) acknowledge problems with large scale document collections. Qiu and Frei (1993) acknowledge that construction of a similarity thesaurus based on millions of documents can be computationally expensive, while Grootjen and van der Weide (2006) report that it is not feasible to calculate a large scale global concept lattice. However, the approach by Grootjen and van der Weide (2006) does not require to calculate the entire lattice, only a sub-lattice with respect to the current query is needed.

In Rajapakse and Denham (2006) each document and query are represented by concept lattices. The concept lattice is a hierarchically ordered conceptual structure based on Formal Concept Analysis (FCA). The concept lattice for a document can be learned and improved by relevance feedback. Testing showed a significant increase in effectiveness as the system learned from experience. They have also recognised the advantage of a hybrid approach where both concepts and keyword matching is used.

Rajapakse and Denham (2002; 2006) have also acknowledged the same problem as (Grootjen & van der Weide, 2006; Qiu & Frei, 1993) with regards to creating a concept lattice for larger document collections. Therefore, they propose to represent each document/query as an individual concept lattice (Rajapakse & Denham, 2006). One advantage is smaller lattices that also allow different weighting of concepts.

3.2.4 Approaches to feature vector construction

In this section, we explore related work on the construction of FVs. FVs can in general be classified as numerical, textual, and hybrid (i.e. a combination of numerical and

textual). In addition, FVs can include other aspects, e.g. properties like category (Sebastiani et al., 2000). In this brief overview, we will focus on approaches using textual FVs, i.e. vectors containing terms with corresponding weights that typically represent a feature like a concept (in Section 4.1 we formally define the Feature Vectors used in this work). Textual FVs are typically based on a lexical resource like WordNet (Lopez et al., 2006b) or extracted from a set of documents (Agirre et al., 2000; Gabrilovich & Markovitch, 2007; Su & Gulla, 2006). There are also approaches that assume FVs are already created (Formica et al., 2008) and consequently focus on the usage of FVs; these approaches will not be considered in this overview. Next, a set of approaches related to our work is analysed.

Explicit Semantic Analysis (ESA) by Gabrilovich and Markovitch (2005; 2007) utilises the vast amount of organised human knowledge that is available in structured repositories like Open Directory Project (ODP). Their feature generation methodology allows the use of external knowledge to construct features. Furthermore, a prerequisite is that these knowledge sources define a collection of concepts with assigned textual documents. An illustrative example of the feature generation process is shown in Figure 3.13. Each node in the ODP is treated as a concept. A textual object is created for each node that consists of concatenated Web documents (listed for each node by ODP) and their textual descriptions (also provided by ODP). The concepts are represented as attribute vectors with their most characteristic words. Since a document can cover diverse topics, they divide each document into non-overlapping segments called contexts. Based on these contexts they generate features. Furthermore, each context is classified into one or several concepts. An ambiguous concept will be part of several domains, which is partly resolved by categorising them. In the case of hierarchies, a parent node also aggregates small fragments of specific knowledge from its ancestors.

They also propose using Wikipedia as an external knowledge source (Gabrilovich & Markovitch, 2009). The underlying approach is similar, but there are some differences. For example, Wikipedia does not have a generalisation hierarchy like ODP, is heavily cross-linked in contrast to ODP, and in general has less content noise compared to ODP. In any case, they report that their approach, independent of using Wikipedia or ODP as an external source, provides significant improvements to current state-of-the-art in automatically assessing the semantic relatedness of texts.

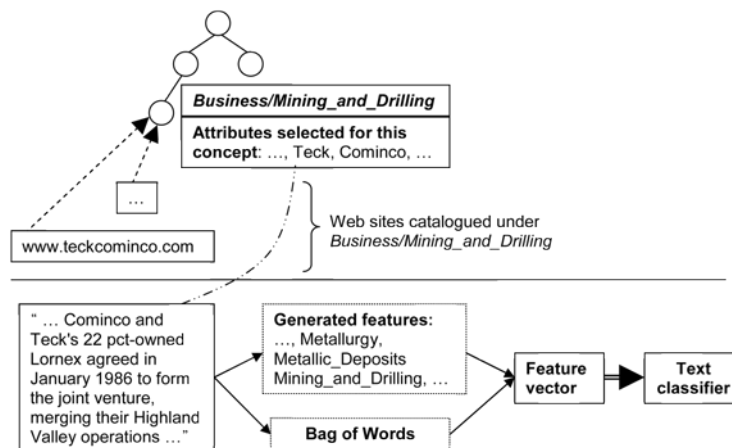


Figure 3.13: An example of generating a feature vector (Gabrilovich & Markovitch, 2005).

Chang et al. (2006) propose an approach to automatic query reformulation called Query Concept Method (QCM). QCM is used to construct query concepts that denote users' information needs. The process of constructing query concepts constitutes of two steps: (1) a set of primitive concepts based on a document collection is constructed; (2) the most associated primitive concepts are selected to be the query concepts. A document collection can either be a text corpus (i.e. a global approach) or a retrieved set of documents (i.e. a local feedback approach). A primitive concept is defined to represent the main topic or meaning of one or more documents. It is represented as a vector of terms that are highly related with the concept in the text (i.e. not necessarily synonyms). Furthermore, primitive concepts are orthogonal within a document and distinct for a corpus. The process of constructing the primitive concepts constitutes of two steps (depicted in Figure 3.14). First, each document is summarised and significant features are extracted. They score each sentence and select the top n ranked sentences for each document. Overlapping sentences are merged. The result of this summarisation and extraction process is a set of orthogonal feature vectors for each document. Then, they generate the primitive concepts by clustering the feature vectors. Ambiguity is handled by applying a classification method using the Yahoo! Directory prior to the clustering of features. Experiments showed promising results, especially for short or poorly formulated queries.

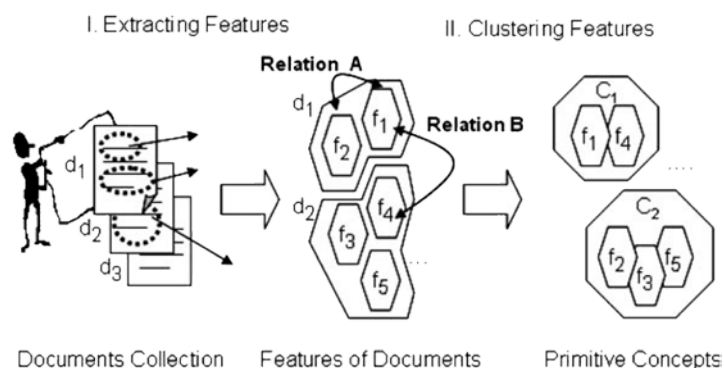


Figure 3.14: The process of constructing primitive concepts (Chang et al., 2006).

Agirre et al. (2000) propose to enrich concepts in existing ontologies, like WordNet (Fellbaum, 1998), using the Web. They propose to associate every concept with a topic signature (i.e. a list of topically related words). A lexicalised concept in WordNet is represented by one or more synonyms called a synset, while each meaning of the concept is called a sense. The process of constructing the topic signatures is depicted in Figure 3.15. First, for each sense, they create a highly specific query that consists of words related to each sense called cue-words (e.g. hyponyms, hypernyms, holonyms). They assume that a document that contains a high number of cue-words that surrounds a target word is likely to correspond to the target concept. The queries created are submitted to a search engine. They retrieved the top 100 documents for each submitted query. They extracted keywords from each document and created a vector of keywords for each collection of documents. Each collection of documents represents a sense of the target word and, hence, the vector represents the corresponding topic signature.

Experiments showed that their approach scored well above the baseline (i.e. choosing senses at random), but in some cases it scored below baseline due to noise. They believe

that formulating the queries was the weakest point of their approach since the quality of the queries highly affected the quality of the documents retrieved. They also experimented with binary hierarchical clustering and various distance metrics, but did not get any substantially different results.

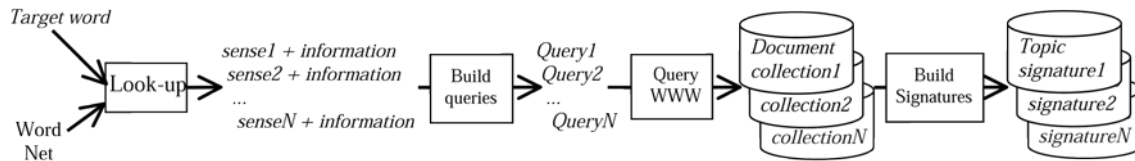


Figure 3.15: The topic signature construction process (Agirre et al., 2000).

Solskinnsbakk and Gulla (2008b) constructed ontological profiles where the ontology classes are represented as concept vectors. Ontological profiles are used to enhance the search. When constructing the concept vectors (depicted in Figure 3.16) they rely on a highly relevant document collection. When indexing the document collection they create three separate indexes (i.e. whole documents, paragraphs and sentences). The ontological profiles are constructed by first finding a set of relevant text elements (i.e. whole documents, paragraphs or sentences) for each ontology concept. The relevant text elements are found by search. Each concept is assigned all terms of the corresponding set of text elements. The final step in the construction process is to calculate the weights for all the terms assigned to each concept. Different indexes are used to boost the weights. Terms that occur in the same sentence as a concept are given higher weights than terms that occur in the same paragraph as a concept, etc. The result is a vector of terms with different weights that reflect the semantic neighbourhood for each concept. Furthermore, they use a collection of irrelevant documents in order to construct negative concept vectors. The concept vectors and their negative vectors are used in query expansion. Testing shows good results for situations where recall is more critical than precision.

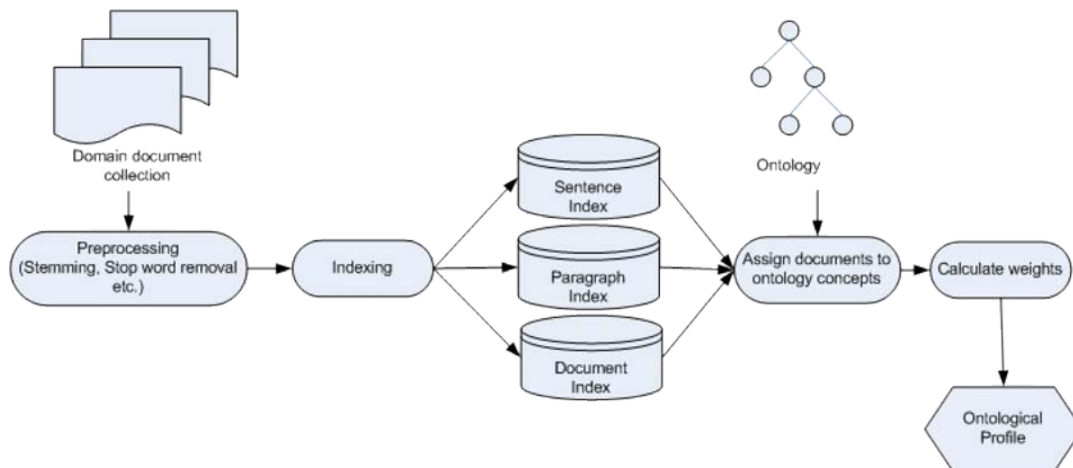


Figure 3.16: The ontological profile construction process (Solskinnsbakk & Gulla, 2008a).

Su and Gulla (2006) propose an ontology mapping approach based on semantic enrichment of ontologies. The ontologies are enriched by assigning a feature vector (FV) to all of their concepts. A FV is constructed from a set of relevant documents (the construction process is depicted in Figure 3.17). They assume a collection of documents

relevant to the ontologies. First, a set of documents is assigned semi-automatically (i.e. users need to adjust the assignments) to each ontology concept using a linguistically based classifier by Brasethvik and Gulla (2001). Then they use the Rocchio classifier algorithm (Manning et al., 2008, p. 269) to construct the FVs. For a leaf concept that does not have any sub-concept, the FV is calculated as the average document vector for all the assigned documents. For a non-leaf concept, the neighbouring concepts are also considered when constructing the FV. In addition to the average document vector for all the assigned documents, it also includes direct sub-concepts and all other directly related concepts. Consequently, a FV for a non-leaf concept is constructed from the three parts which are individually weighted. The result is an ontology with a FV assigned to each of its concepts. Next, the enriched ontologies are used to map concepts. They report that the approach was capable of finding most of the mappings, and ranking them correctly.

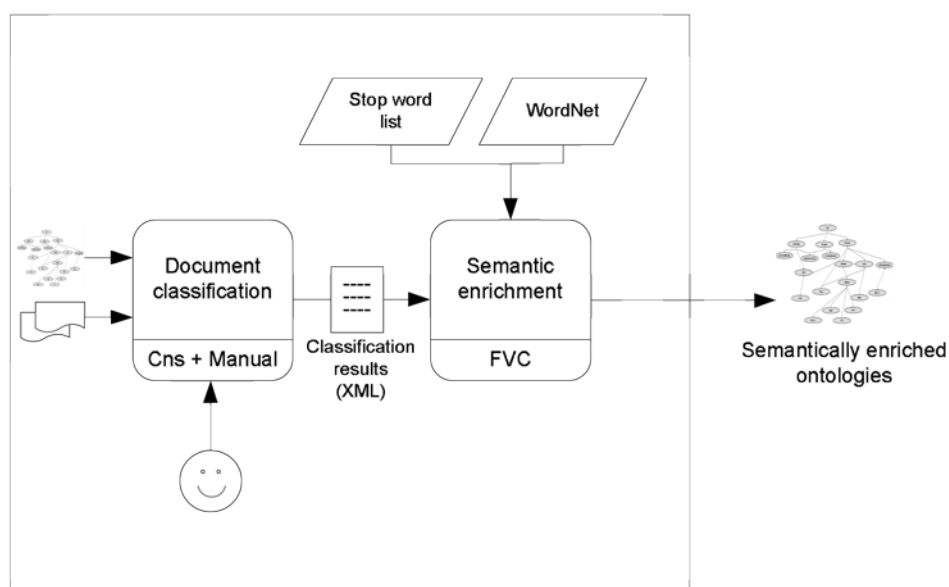


Figure 3.17: The proposed semantic enrichment process (Su & Gulla, 2006).

Kulkarni and Caragea (2009) propose a Concept Extractor and Relationship Identifier (CE-RI) system to bridge the gap between the current Web and the Semantic Web. The approach does not use feature vectors or anything similar. The Concept Extractor (CE) component (depicted on the left in Figure 3.18) extracts concepts related to a user's query, while the Relationship Identifier (RI) component (depicted on the right in Figure 3.18) finds relationships between the extracted concepts and the user's query. The search results are presented to the user as a Semantic Relationship Graph. The CE component utilises the power of existing search engines to collect sets of relevant documents with respect to a reformulated user query (i.e. they consider all possible keyword combinations). The reformulated queries are sent to a search engine; resulting document links are extracted and used to create a local document corpus. Then PageRank (Page et al., 1999) is used to find the most representative documents. Based on the top n documents, they extract concepts. When extracting the concepts they rely on meta information being available in the documents, more specifically, meta keywords and the titles of the Web pages. Finally, they calculate a weight for each extracted concept. Next, the RI component identifies the relationships between query

concepts and concepts extracted by the CE component. First, it uses WordNet to find a relationship. If no relationship is found on WordNet, then Wikipedia is used and, finally, Yahoo! Directory if Wikipedia should also fail. The final result is stored in a RDF database and presented to the user as a Semantic Relationship Graph. They report that the approach can capture loose relationships but struggles with more exact relationships. In any case, experiments show similar or better results than comparable systems.

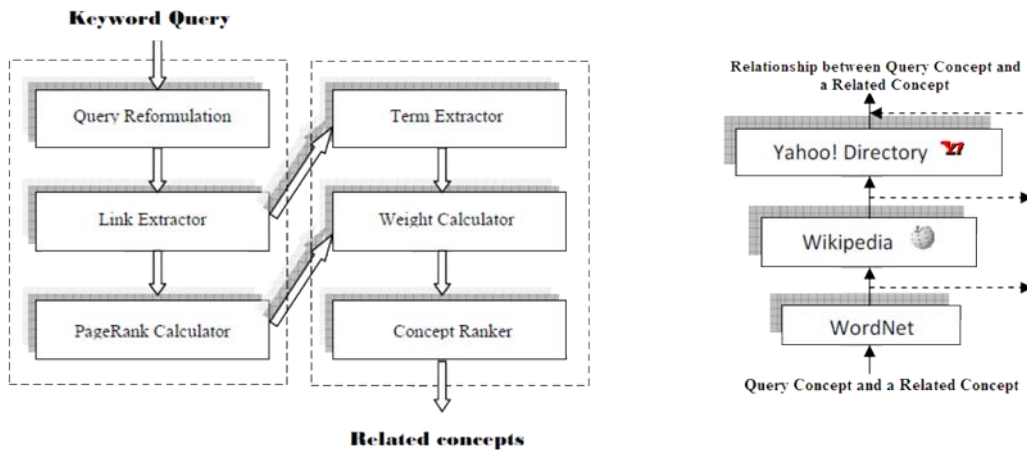


Figure 3.18: The architecture of CE (left) and RI (right) (Kulkarni & Caragea, 2009).

3.3 Evaluation of semantic search systems

Existing evaluations of semantic search systems report improvements compared to traditional search systems (see Section 3.2). The majority of the systems reported in Section 3.2.1 were evaluated as black boxes (i.e. measuring only the output). However, semantic search systems typically add extra complexity to user interaction that is ignored by the traditional measures of effectiveness like precision and recall. The ultimate goal of IR evaluation is to assess the probability of an IR system being both adopted and used. Consequently, users' satisfaction and other measures also need to be considered when evaluating the success of such systems. In general, approaches to information retrieval systems evaluation can be divided into system- and user-centric approaches. In addition, according to Borlund (2009), there is also a third category; cognitive IR evaluation approaches that views and treats the system and users as a whole. However, in this brief overview we will focus on the former two approaches to evaluation of information retrieval systems rather than the latter. Furthermore, we will have a special focus on approaches to evaluation of semantic search systems in particular.

System-centric evaluation methods typically assess retrieved information by its relevance to the users. The information retrieval systems are compared based on their ability to retrieve and rank relevant information. The traditional measures of effectiveness are precision and recall (Baeza-Yates & Ribeiro-Neto, 1999, p. 75). From these traditional measures several similar metrics have been derived such as novelty, coverage, the E-measure, Harmonic mean (a.k.a. F-measure) (Baeza-Yates & Ribeiro-Neto, 1999, p. 82). An overview of efficiency based metrics is provided in (Demartini & Mizzaro, 2006).

Precision and recall measures, and similar, are used to compare results by retrieving information from standard document collections like Text REtrieval Conference (TREC). The TREC initiative was established to provide a testbed for information retrieval systems (Voorhees & Harman, 2005). TREC manages document sets, topic (or query) sets, relevance judgments provided by domain experts, tasks, and tracks to evaluate different aspects of an IR system. However, based on these traditional measures it is difficult to find the causes for variations of different retrieval results (Alemayehu, 2003).

It is widely accepted that external factors exist that can affect the retrieval results considerably (cf., Alemayehu, 2003; Gao et al., 2004; Harter, 1996). Alemayehu (2003) and Gao et al. (2004) argue that factors like indexing and searching methods, familiarity of search topic, and query quality also need to be considered in the evaluation of a system. Gao et al. (2004) propose a two-dimensional evaluation of retrieval results (i.e. IR system and IR environment) to identify potential problems affecting retrieval effectiveness. Knowledge about factors affecting retrieval effectiveness can help to noticeably improve the retrieval approach (Alemayehu, 2003).

System-centric evaluation approaches are convenient for comparing search engines over time, since earlier results can easily be compared with newer results. However, these evaluation approaches do not correlate well with users' perceptions of success (Harter, 1992; Huffman & Hochster, 2007; Spink, 2002; Su, 1992; Wang & Forgionne, 2008). According to Harter (1992):

"For relevance judgments are a function of one's mental state at the time a reference is read. They are not fixed; they are dynamic. Recording such judgments, treating them as permanent, unchanging relations between a document set and a question set, and then using them to compute such measures as recall and precision to evaluate retrieval effectiveness, is contrary to the meaning of psychological relevance."

Harter quite clearly states that evaluation of IR systems using a fixed set of documents and queries does not reflect reality, and hence provides a limited insight when evaluating the potential success of a system.

User-centric evaluation approaches look upon IR systems in a much broader way than system-centric evaluation approaches, viewing the system as whole. The user is more involved in the evaluation of the system (e.g. the relevance judgment is given by the user with respect to his or her information need). Su (1992) found that users' satisfaction with completeness of search results and value of search results as a whole, among other measures, were significantly correlated with success. Therefore, in some evaluations user satisfaction is equated with system effectiveness. But the downside is that user-centric evaluation approaches are less scalable and repeatable when compared to system-centric evaluation approaches (Huffman & Hochster, 2007).

In a broader perspective, Venkatesh et al. (2003) have identified various factors that affect user satisfaction, like performance expectancy, effort expectancy, experience and voluntariness of use. While DeLone and McLean (2003) propose a model containing six main dimensions for categorising different measures of information system success: system quality, information quality, service quality, use (i.e. both intention to use and use), user satisfaction, and net benefits. They articulate that system, service and

information qualities singularly and jointly affect both the intention to use and user satisfaction. Additionally, the amount of use can affect the degree of user satisfaction as well as the reverse being true. Use and user satisfaction directly impact on net benefits that may be achieved from the information system (Delone & McLean, 2003).

Wang and Forgionne (2008) propose a framework where efficiency is related to the time needed for a user to perform each decision-making step during the IR process (e.g. the time to recognise the problem, establish search queries, and to identify relevant documents). They relate effectiveness to a user's decision productivity at each step during the retrieval process (e.g. the number of general topic alternatives and the number of relevant documents identified). Griffiths et al. (2007) analyse four factors that affect a user's satisfaction, namely, system, user, environmental, and task factors. Consequently, they define user satisfaction as a complex construct consisting of the output of the search, the view of the system as a whole (i.e. its features and functionality), and the user's whole experience in interacting with the system via its interface.

In Section 3.1, we categorised aspects of semantic search systems. As has been shown, there is a diversity of user interfaces adopted by semantic search systems. Consequently, an important aspect is to include the end-user in all search phases when evaluating such systems. For instance, approaches based on structured ontology query languages (Blacoe et al., 2008; Wang et al., 2008) require advanced knowledge of designated query languages and best suit professional knowledge users. Contrarily, approaches based on graphical ontology browsing (Brasethvik, 2004; Suomela & Kekalainen, 2005) typically targets novice users, but larger ontologies can still be complicated to use (Nagypal, 2005). Still, few evaluations of semantic search systems include profiling of the participants (Castells et al., 2007; Solskinnsbakk & Gulla, 2008b; Wang et al., 2008).

Moreover, in a typical evaluation, a limited amount of queries are used and the queries are often pre-formulated by the researchers (Jiang & Tan, 2006; Rocha et al., 2004). Consequently, the evaluations are often conducted without any involvement of end-users (Borlund, 2009; Schumacher et al., 2008; Wang et al., 2008). Normally, the end-users only evaluate retrieved results with respect to simulated information needs (Formica et al., 2008; Solskinnsbakk & Gulla, 2008b). However, there is a difference in who makes the judgement. For example, Paralic and Kostial (2003) leave the relevance judgments to domain experts of Cystic Fibrosis while Jiang and Tan (2006) rely on relevance judgments by research colleagues.

Furthermore, there is a tendency to barely focus on performance measures, i.e. precision and recall figures (Castells et al., 2007; Zhang et al., 2005). However there are some exceptions. For instance, McCool et al. (2005) tested the TAP search engine (Guha et al., 2003) by measuring user satisfaction of interaction with the system, but they did not measure the quality of the retrieved documents.

In Table 3.3, we summarise the evaluations of the approaches reviewed in Section 3.2.1. As can be seen from the table, most of the evaluations were system-centric while only three were categorised as user-centric. For most of the system-centric evaluations, humans judged the relevance of the retrieved results because there are few ontologies that cover the standard test collection's queries. Only two approaches were evaluated

using standardised document collections, while Nagypal (2005; 2007) evaluated his system using Wikipedia. Two of the user-centric evaluations were categorised as partially user-centric (cf., Lopez et al., 2007; Zhang et al., 2005) since the users could formulate their own queries based on simulated information needs, but only the retrieved results were evaluated. The evaluation by McCool et al. (2005) was the only evaluation that was classified as user-centric since they measured the users' satisfaction though the quality of the retrieved results was not explicitly evaluated.

Table 3.3: Summary of evaluation approaches.

	<i>System-centric</i>		<i>Partially user-centric</i>	<i>User-centric</i>	<i>Comment</i>
	<i>Automation</i>	<i>Human</i>			
Blacoe et al. (2008)		X			
Braga et al. (2000)					No evaluation
Castells et al. (2007)		X			
Formica et al. (2008)		X			
Guha et al. (2003)					Evaluated by (McCool et al., 2005)
Harth et al. (2007)					No evaluation
Jiang and Tan (2006)		X			
Lopez et al. (2006a; 2006b)					No evaluation
Lopez et al. (2007)			X		Only search result quality
McCool et al. (2005)				X	
Nagypal (2005; 2007)	X				Based on Wikipedia
Paralic and Kostial (2003)	X				Cystic Fibrosis
Rocha et al. (2004)		X			
Schumacher et al. (2008)		X			
Solskinnsbakk and Gulla (2008b)		X			
Wang et al. (2008)		X			
Zhang et al. (2005)		X	X		Conducted two evaluations
Zhou et al. (2007)	X				TREC

Note: Automation and human refer to relevance judgment approaches.

There are in general two main objectives when evaluating semantic search systems. The first is to prove their advantage over existing search engines while the second is to assess its potential usage. Aiming at the first objective, using a static document collection like TREC could be a natural choice. However, this is still problematic since online ontologies only cover a fraction of test collections' queries (d'Aquin et al., 2008; Jiang & Tan, 2006; Nagypal, 2005). In Section 3.2.1, we reviewed a set of approaches, their evaluations are summarised in Table 3.3. As can be observed from Table 3.3, only two (note that (Nagypal, 2005; 2007) created own test collection) of these approaches (cf., Paralic & Kostial, 2003; Zhou et al., 2007) used an already established fixed set of documents and queries when evaluating their systems. In most of these evaluations, their own data sets were used and focused mainly on performance issues like precision and recall. In these evaluations, little attention was given to the second evaluation objective, to assess its potential usage. Only three of the reviewed approaches included

users in the evaluation process for more than only assessing the relevance of the search results. Since the ultimate success factor for each search systems is end-users' satisfaction and given that semantic search systems tend to add extra complexity to user interaction, potential end-users need to be more involved (e.g. formulate own queries) when evaluating the retrieval effectiveness of a semantic search system. The evaluation approaches used in this work is described in Section 4.3.

3.4 Summary

The overview of related work provided in this chapter covers semantic search and approaches to feature vector construction that are relevant to this research. Their relevance can be summarised as follows (a more extensive analysis with respect to the contributions is in Section 5.3):

- **Academic approaches** to semantic search are mainly based on semantic annotation of documents. Typically, whole documents are treated as ontology instances. Consequently, this results in a coarse retrieval of information since they focus on retrieving instances rather than documents.
- **Commercial approaches** to semantic search tend to combine information from a variety of different sources into a united view of retrieved results. Most approaches focus on categorisation of results and try to avoid the traditional listing of search results. Since limited information is available about these systems, it was difficult to find out whether ontologies were used. However, two of the reviewed systems (True Knowledge and Powerset) target question answering and both represent underlying knowledge as triples.
- **Query reformulation approaches** mainly focus on creating concepts based on either a global (i.e. text corpus) or a local (i.e. query result set) document set. Several have acknowledged negative performance issues with respect to creating concepts for larger documents collections. Therefore, many approaches focus on local document sets instead.
- **Approaches to feature vector construction** are sparse, especially approaches utilising ontologies. Many approaches assume that highly relevant documents are assigned to the ontology concepts and from there create the FVs and hence mainly circumvent the difficulty of word sense disambiguation.
- **Evaluations of semantic search systems** are mainly system-centred. Humans often judge the quality of the retrieved results since there are few ontologies that cover standardised test collections. System-centric evaluations are incapable of reflecting the added complexity of semantic search systems and that relevance is not static but multidimensional. Furthermore, the end-users are seldom involved. Consequently, it is difficult to assess the potential usefulness of the evaluated approach.

In Section 1.4, a set of research questions was formulated. With regard to research question 1 (RQ1), based on the related work reviewed we can state that retrieval effectiveness can be improved by utilising ontologies. However, we also found that the majority of the systems are evaluated as black boxes (see Section 3.3) - that is, by measuring only the output using traditional precision and recall measures (i.e. internal components and interaction between them is ignored). Typically, the added complexity of using semantic search systems is also ignored.

With respect to RQ2, we found that there is variety of approaches capable of relating terminologies provided in ontologies with textual documents and queries (i.e. mainly based on a one-to-one match of lexical terms). However, we could not find any approach extracting terms that are used in connection with concepts defined by ontologies. Consequently, we neither found an approach to how these associations could be evaluated nor which features of an ontology are most useful and yield the best results (see RQ3 and RQ4 in Section 1.4).

4 Results

This chapter summarises the main results of this work. First, we present and define feature vectors since they are the cornerstones to this work. Then, we give an overview of implemented prototypes before summarising the experiments conducted. Finally, we provide structured abstracts for each of the published results. More details are in papers P1-P8 (Part II of this thesis).

4.1 Feature Vectors

The development of our approach is inspired by a linguistic method for describing meaning of objects - the *triangle of reference*, also known as the semiotic triangle by Ogden and Richards (1927, p. 11). Ogden and Richards describe how symbols are connected to referents (i.e. objects), not directly but only indirectly through thoughts or references. Equally, in our approach a Feature Vector (FV) "connects" an entity, encoded in an ontology, to the actual terminology used in a document collection (see Figure 4.1). A FV is built considering both the semantics (i.e. the semantic neighbourhood of an entity) encoded in an ontology and the dominant lexical terminology surrounding the entity (i.e. linguistic neighbourhood) in a text corpus. The semantic neighbourhood is computed based on related entities and direct properties specified in an ontology, while the linguistic neighbourhood is based on the co-location of terms in a document collection. Therefore, a FV reflects both the semantic and linguistic neighbourhoods of a particular entity and hence constitutes a rich representation of an entity that is related to the actual terminology used in a text corpus. Figure 4.1 shows an illustration of a FV and how it relates to an entity and a set of documents.

A FV of an entity e is represented as a two-tuple and is defined as follows:

Definition: *Feature Vector (FV)*

$$FV_e = \langle S_e, L_e \rangle \mid S_e \in O_d, L_e \in D_d$$

$$S_e = (e_i, DR_{e_i})$$

$$DR_{e_i} = Parents_{e_i} \cup Children_{e_i} \cup Others_{e_i} = \{ \langle e_i, e_k \rangle \} \subseteq E \times E$$

$$L_{e_i} = collocated(S_{e_i}, L_{e_{Dd}})$$

where S_e is a semantic enrichment part of FV_e that represents a set of neighbourhood entities and properties in an ontology O of a domain d . L_e is a linguistic enrichment of a entity that is a set of terms (from document collection D of a particular domain d) with a significant proximity to an entity and its semantic neighbourhood.

Jaguar = {jaguar (0,54), species (0,09), cat (0,09), range (0,07), prey (0,06),
habitat (0,04), culture (0,04), panthera onca (0,04), population (0,04)}

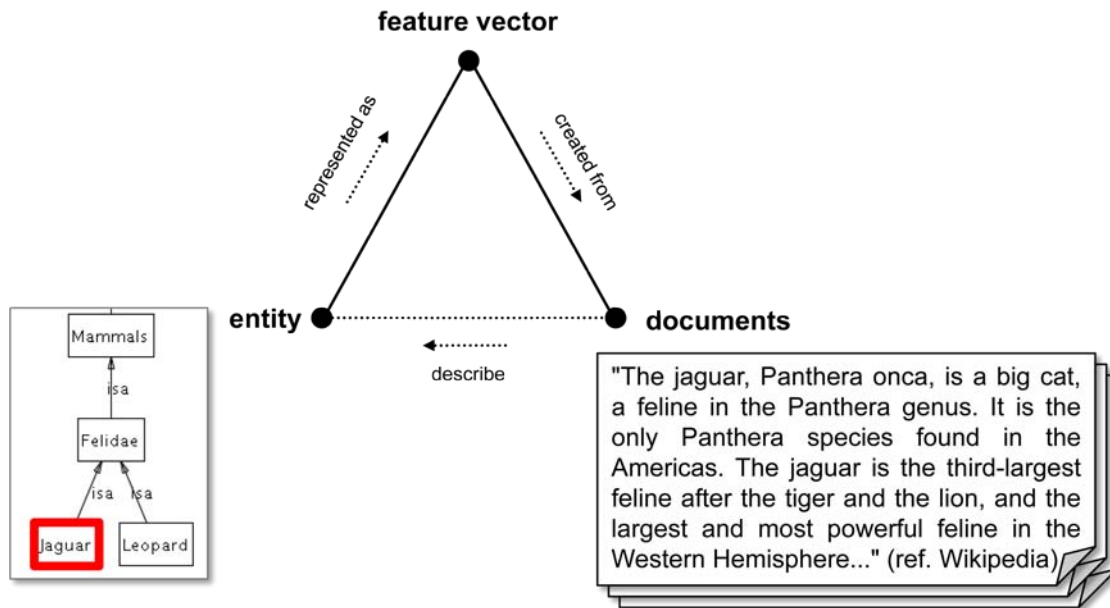


Figure 4.1: An illustration of the relationship between a feature vector, an entity and a set of documents.

4.2 Implementations

In this section, we describe the prototypes implemented as part of this work. First, we provide an overview of the semantic search system. Then, we go into more details of the two prototypes, emphasising on the differences between them. The first prototype (Prototype I), was implemented to test FVs used to disambiguate search, while the second prototype (Prototype II) to test an alternative feature vector construction algorithm than that used in the first prototype.

The overall architecture of the implemented system was created in the analysis and design phase of this research phase (Phase I: Analysis and design, Section 2.3.1). Prototype I and II were implemented as part of the two consecutive research phases (Phase II and III described in Section 2.3.1). Next, the overall architecture and the two prototypes are presented.

4.2.1 Analysis and design

The overall architecture of the ontology-driven information retrieval system is depicted in Figure 4.2. Prototype I and II have the same overall architecture. The main difference between the prototypes is the feature vector miner component, where each prototype uses a different feature vector construction algorithm.

As seen in Figure 4.2, the system is made of a set of offline and online components⁶. In addition, there are some components that are used both offline and online. A brief

⁶ By offline and online we differentiate component usage with respect to the search session itself.

description of the individual components is provided, before the prototypes are presented in the following subsections.

Feature vector miner (offline): This component takes an ontology, from the *ontology repository*, and automatically creates FVs for each of the ontology entities. The FVs are stored in the *feature vector repository*.

Ontology-driven retrieval engine (online): This component performs semantic search where the FVs created, from the *feature vector repository*, are used to disambiguated query terms.

Query and indexing system wrappers (offline and online): This component creates a common interface to the various query and indexing systems (i.e. search engines).

Ontology Repository (offline and online): This repository contains the ontologies used.

Feature Vector Repository (offline and online): Contains the feature vectors of the corresponding ontology entities found in the *ontology repository*.

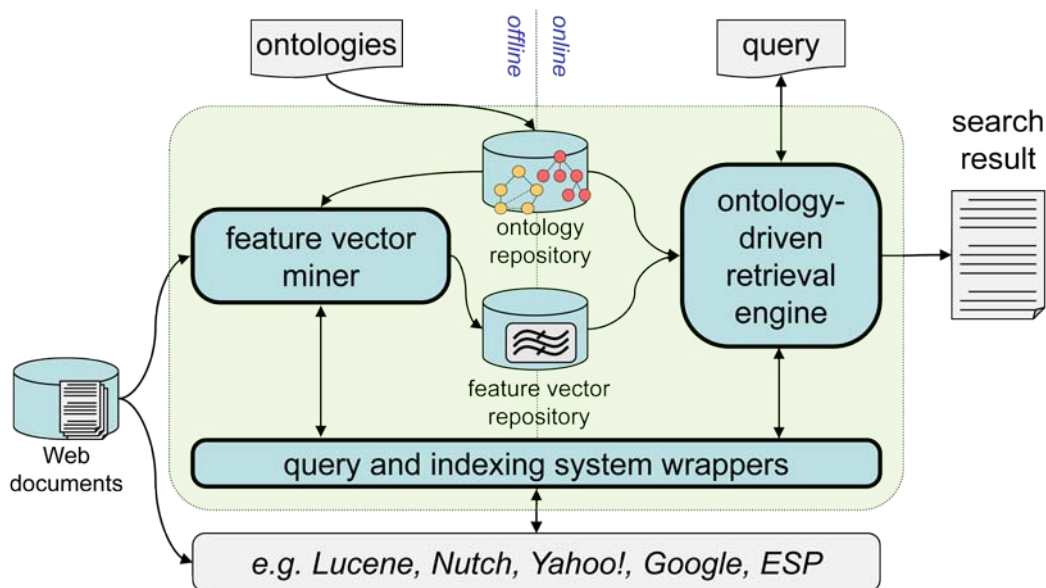


Figure 4.2: The architecture of the ontology-driven information retrieval system.

4.2.2 Prototype I

This first prototype included a semantic search system (the overall architecture of this system is depicted in Figure 4.2) that performed as an extension of an existing search system (e.g. Yahoo!). The general idea was to extend an existing search engine with semantic capabilities (i.e. use ontologies to disambiguate search). The prototype was implemented in Java and contained more than 18,000 lines of code (LOC). It was designed as a Web application and ran on an Apache Tomcat (Apache, 1999) server.

This first prototype was evaluated by real users (Experiment I is presented in Section 4.3). One goal for this prototype, and hence for the experiment, was to create a close to real life search situation. Consequently, the intention was to create a Web user interface similar to a typical search engine found on the Web (see Figure 4.3) to make the interface as familiar as possible for users. Typically users type in their query in a single text field (our *Terms* field depicted in Figure 4.3 provides the same functionality). In

addition, we required (in this version of the prototype) the users to specify both a domain of interest and one or more concepts (i.e. entity) within the selected domain. To assist the user in finding appropriate concepts a suggest-like interface was implemented (i.e. when the user started typing, a list of selectable entity names was suggested). A set of simulated information needs used in the experiment (described in Section 4.3) helped the user select appropriate domains and concepts. More details are provided in paper P6.

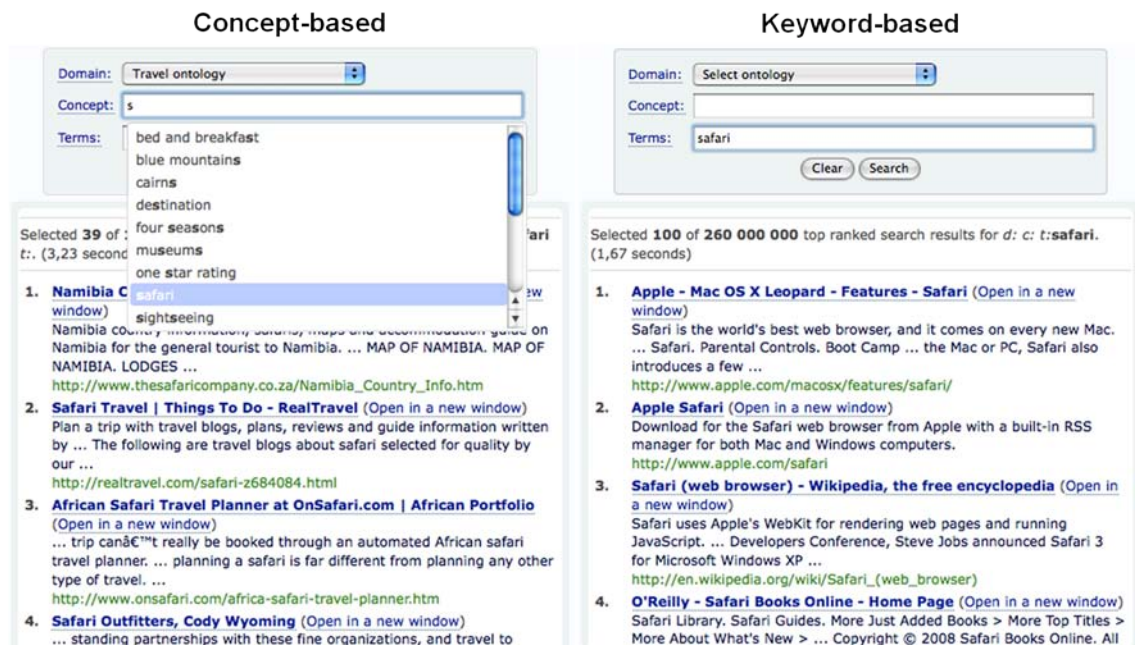


Figure 4.3: The search user interface of Prototype I.

Figure 4.4 depicts an overview of the different steps of the search process. First, a user needs to formulate a *query*. The user can specify one or more entities related to the domain of interest. In addition, the user can specify a set of keywords to narrow the search even further (see Figure 4.3). By differentiating between entities and keywords, the real intention of a user's query can be interpreted better by the underlying machinery and thus present more relevant results. The *query* is submitted to the *ontology-driven retrieval engine* that identifies the corresponding entities of the ontologies and submits a semantically enriched query to the underlying *query and indexing system*. Query terms with no corresponding entities are treated as ordinary keywords. Then, the *ontology-driven retrieval engine* creates a document feature vector (DFV) for each document in the search result by the *query and indexing system*. Then the DFVs are compared with the entities (i.e. their corresponding FVs) specified in the user's query. Those documents having a similarity score below a specified threshold are disregarded, while the other documents are re-ranked according to the similarity scores and presented to the user.

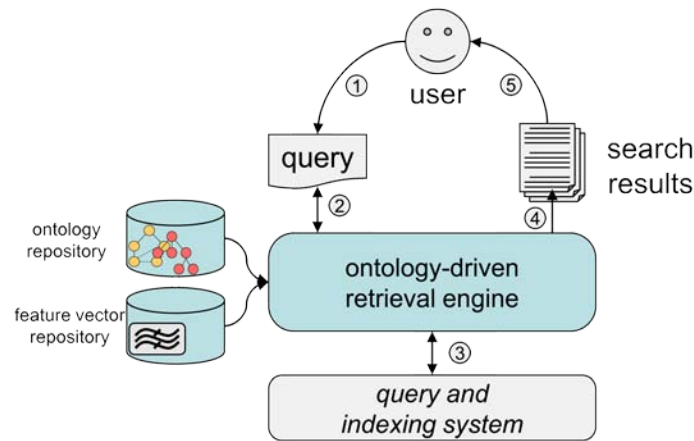


Figure 4.4: An overview of the search process.

The main difference between Prototype I and II is the *feature vector miner* component that has different Feature Vector Construction (FVC) algorithms. An overview of the algorithm used by the first prototype is depicted in Figure 4.5 (the left side of the figure) along with an illustration of the construction process (the right side of the figure). As can be seen from Figure 4.5 the algorithm constitutes two main steps (each main step includes a set of sub steps). The first main step aims to extract and group candidate terms relevant to each entity (the Lingo clustering algorithm part of the Carrot² framework (Carrot2, 2009) was used). However, not all the candidate terms are necessarily relevant to the domain described by the ontology. Consequently, the aim of the last main step is to identify the most relevant group of candidate terms with respect to the ontology. This is done by comparing all the candidate groups of an entity with all the candidate groups of neighbouring entities. Finally, an FV for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding FVs that consist of terms associated to both the entities (from the ontologies) and the domain terminology (from the text corpus). More details of this algorithm are provided in papers P1 and P2.

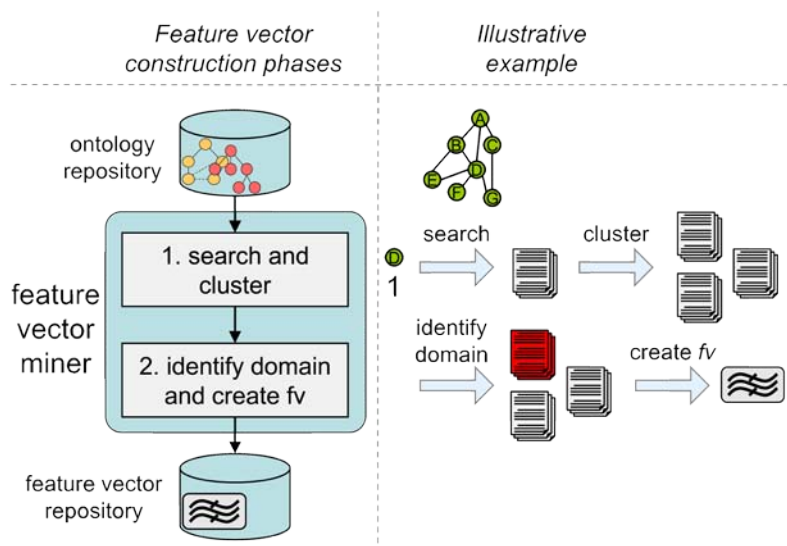


Figure 4.5: Overview of the first feature vector construction algorithm.

4.2.3 Prototype II

The second prototype was based on the first prototype. The feature vector miner component was redesigned and re-implemented based on lessons learned from the first prototype, resulting in a new FV construction (FVC) algorithm. More than 2.000 LOC were added and more than 1.500 LOC were re-implemented, totalling more than 20.000 LOC.

The main difference to the first prototype is the new FVC algorithm (depicted in Figure 4.6). The FVC process is composed of three main steps, while the FVC process presented in Figure 4.5 contains two main steps. The first main step is entirely new and includes ranking of the entities of an ontology according to their assumed importance (i.e. centrality) with respect to the ontology. In Prototype I, the entities were unordered. The main aim of second step is to extract candidate terms from a text corpus relevant to each entity and group these into sets of candidate terms (the Suffix Tree Clustering (STC) algorithm part of the Carrot² framework (Carrot2, 2009) was used). At this stage, the ontology entities are treated as terms and hence can be of any domain (i.e. homonyms). Consequently, the aim of the third main step (this main step is also new because it is dependent on the ranked list of entities created in the first step) is to identify those sets of candidate terms most relevant to the entities defined by the ontology. Finally, a FV for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding FVs that consists of terms associated to both the entities and the domain terminology.

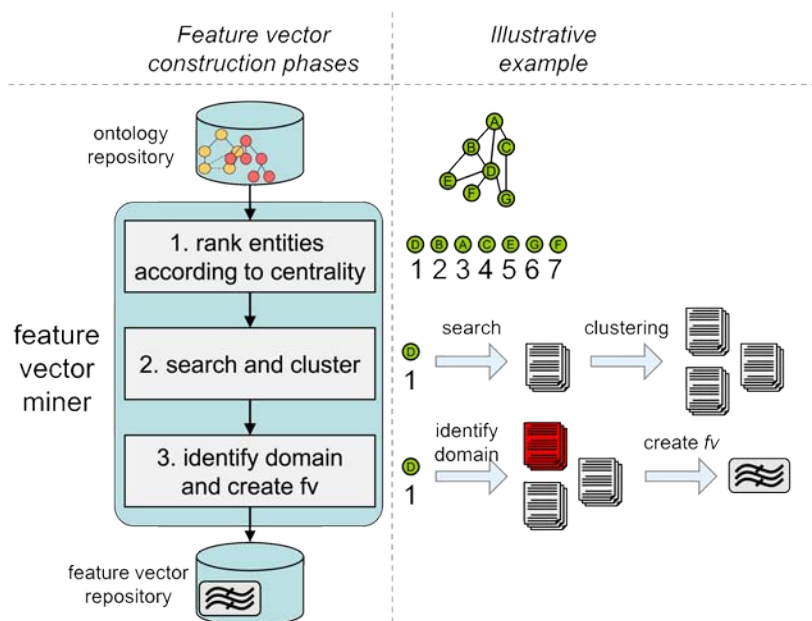


Figure 4.6: Overview of the second feature vector construction algorithm.

Bear in mind, that the first main step included ranking of the entities prior to associating relevant terms. This list of ranked entities is later used in main step three to identify those candidate terms being most relevant to the domain defined by the ontology. The hypothesis here was that a ranked list in contrast to a random list of entities, as used in

Prototype I, would improve the quality of identifying the most relevant candidate terms (executed in Step 3). The idea was that more information (i.e. knowledge that can be extracted from the ontology about an entity) is available for the most central entities and are hence better candidates to discriminate the most relevant terms with respect to the ontology. Therefore, already processed entities (i.e. those already assigned FVs) are used in the process to identify the most relevant terms for the unprocessed entities and so forth. More details about the algorithm can be found in paper P3, while in paper P4 a more extensive experiment is conducted to test aspects of both algorithms.

4.3 Experiments

In this subsection, we provide general information about the experiments conducted as part of this work. First, we present information relevant to both experiments before we introduce each of the experiments in detail. More details about the results and evaluations of the experiments are given in the extended abstracts and the papers P1-P6.

In Experiment I, we evaluated the retrieval effectiveness (see our definition in Section 1.2) of Prototype I. The participants were involved in the evaluation process. The participants formulated their own queries based on information needs provided, assessed the relevance of retrieved results, and completed a post-task questionnaire. In the evaluation of the results, the queries, corresponding relevance scores, and the questionnaires were used. While in Experiment II, we evaluated the sensitivity of the feature vector construction approach. This evaluation was conducted in a laboratory setting. Finally, in Experiment III, we validated our results from Experiment II with real users. However, this was a controlled experiment where the participants performed human judgement on retrieved results.

The experiments were all performed on a standard PC with an Intel[®] Pentium processor running Windows[®] XP and Apache Tomcat. In Experiment I, the Yahoo![®] Web Search API (Yahoo, 2009) was used, while in Experiment II and III the Google[®] AJAX Search API (Google, 2009) was also used.

A set of ontologies was used throughout the experiments (see Appendix I). The selected ontologies are of different granularity that can generally be divided into three categories: taxonomy, lightweight, and advanced. Furthermore, we decided to exclude heavyweight ontologies (i.e. advanced ontologies with several thousand entities) since larger ontologies were not believed to provide any significantly new insight except with regards to processing time, which was not the focus of this work. The reason for this is the nature of the feature vector construction algorithms, i.e. they are locally oriented with respect to the entities (more information regarding these algorithms is provided in papers P2 and P3). All the ontologies were formalised in OWL (McGuinness & van Harmelen, 2004).

4.3.1 Experiment I

Objectives

The objective of this experiment was to evaluate our proposed approach to semantic search built on the concept of feature vectors (FVs). The experiment was conducted with potential end-users of such systems. In this experiment, we evaluate the quality of

the FVs indirectly by their search performance. We analysed the sensitivity of the approach with respect to ontology quality and search tasks. In addition, a post-task questionnaire was used to evaluate the approach.

Settings

The overall design for this experiment is depicted in Figure 4.7. The test subjects were given eight topics and descriptions of simulated information needs (listed in Appendix E) from four different domains. They needed to formulate a total of 16 queries each (eight submitted to the prototype and eight to the baseline). The queries were specified using keywords and entities from particular domain ontologies. The system returned 10 top ranked documents for each query. Each document was assessed by a perceived relevance. After completing the experiment, the test subjects completed a questionnaire of 29 questions (the questionnaire is in Appendix F and the results are in Appendix G).

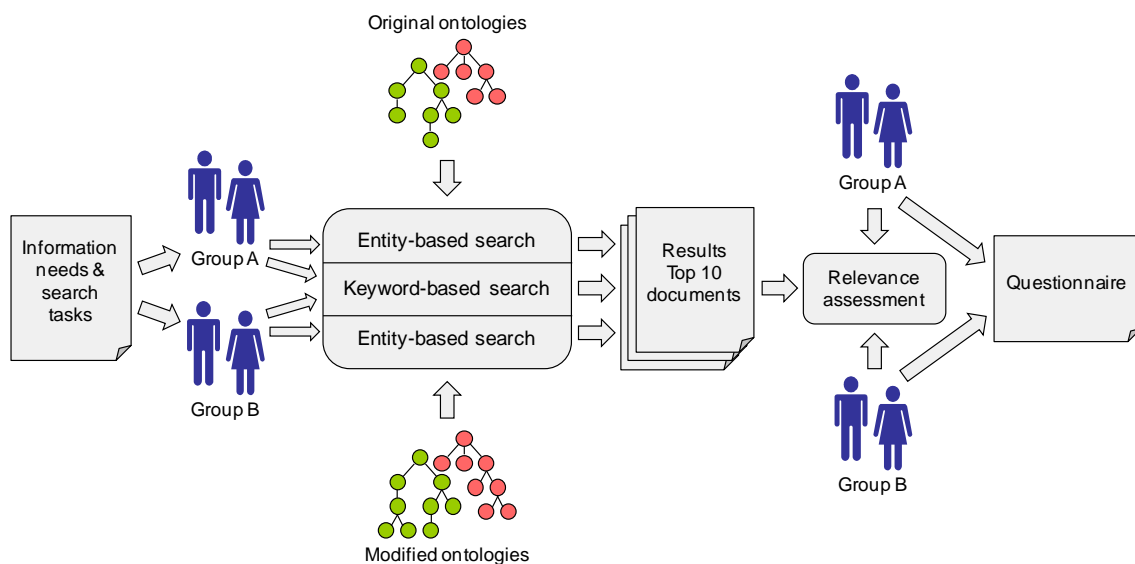


Figure 4.7: Design of Experiment I.

The test subjects were divided into two groups that used different ontologies for the same domain (see Figure 4.7). The first group used the original ontology while the second group used an altered version of the original ontology. The original ontology was altered to include more relationships and/or instances to see if this would influence the search results. Ontologies were modified by adding instances (all ontologies), specifying additional object properties (travel, animal and wine ontologies) and refining taxonomical relationships (animal ontology). The results of these changes were different feature vectors generated for the same entities of the two different, but still similar, ontologies. In summary, group 1 contained 10 participants, while group 2 had 11 participants. In total, the users executed 81 queries using the original ontologies and 92 queries using the modified ontologies, and 152 were simple keyword based queries executed directly to the baseline. The ontologies used are listed in Appendix I.

Results and conclusions

The participants were mainly 4th year students at the Norwegian University of Science and Technology (NTNU) (see Table 4.1 for demographic information about the

participants). The 21 test subjects were offered payment for their time after full completion of the experiment. The Yahoo! Web Search was selected as the backend search engine.

Table 4.1: Demographic and background information about the participants.

Demographic feature	Response	Demographic feature	Response	Demographic feature	Response			
Gender	male: 18 (86%)	Amount of keywords in a good query	2 or less	Knowledge about ontologies	None			
	female: 3 (14%)		4		1 (5%)			
			3		11 (52%)	Have heard about	9 (43%)	
Age	[18-24]: 13 (62%)		4	6 (29%)	Have been studying	5 (24%)		
	[25-29]: 5 (24%)		5	0 (0%)	Have been using in prototyping	6 (29%)		
	[30-39]: 2 (9%)		6 or more	0 (0%)	Practical development	0 (0%)		
	[40-49]: 1 (5%)							
Web search experience	None: 0 (0%)	Search service preference	Generic Web search:	20 (95%)	Participation in evaluations	First time:	4 (19%)	
	Sparse: 0 (0%)		Specialized Web search:	5 (24%)		Sparse:	7 (33%)	
	Moderate: 5 (24%)		On-line catalogues:	0 (0%)		Moderate:	8 (38%)	
	Extensive as user:		10 (48%)	Specialized digital libraries:		8 (38%)	Extensive as participant:	1 (5%)
	Extensive as user and developer:		6 (28%)	Other (journal site, wikipedia, google specialised search):		3 (14%)	Both as participant & evaluator:	1 (5%)

Table 4.2 summarises the results of the evaluation with respect to the keyword-based search (i.e. the baseline). From the table we can observe that ontology version 1 performs slightly worse (-0.2%) than the baseline, while ontology version 2 performs better (10.5%).

Table 4.2: Comparison of mean relevance score of keyword and concept based searches.

	Mean relevance score	Diff. from baseline
Keyword-based	42.2	-
Ontology ver. 1	42.1	-0.2%
Ontology ver. 2	46.6	10.5%

Table 4.3 depicts the relevance scores with respect to the different ontology versions. The modified ontologies yielded an improvement in the average score of 10.7%. This indicates that in general a more detailed ontology performs better than a similar, less detailed ontology. More advanced ontologies contain more information that directly contribute to better quality of the entity FVs and hence will contain less noisy terms compared to those created from a basic ontology. One of the reasons for the Travel ontology version 2 performing worse than version 1 is the quality of those ontologies (Strasunskas & Tomassen, 2008a).

Table 4.3: Average relevance scores versus ontology version.

	Ontology ver. 1	Ontology ver. 2	Diff. (%)
Animals	19.4	38.0	96.6%
Autos	32.9	33.7	2.2%
Travel	71.8	65.2	-9.1%
Wine&Food	42.9	51.8	20.6%
<i>Overall</i>	<i>42.1</i>	<i>46.6</i>	<i>10.7%</i>

Table 4.4 depicts the results with respect to each topic (see Appendix E) where we have included related questionnaire items. Topic 3 and 4 provided the best relevance scores for both ontology versions. From Table 4.4 we can observe that the participants were fairly familiar with the retrieval tasks, but still found the ontologies and quality of descriptions good and the presence of concepts in descriptions helpful when formulating

those queries. For the topic with the lowest relevance score, topic 6, we observed the opposite effect. Topic 6 also had the biggest variance in query length, probably due to the increased difficulty in formulating a suitable query for this topic (Strasunskas & Tomassen, 2008b).

Table 4.4: Mean scores on questionnaire items regarding the experiment.

Domain	Topics	Familiarity w/ retrieval tasks	Ontology usefulness	Quality of info needs and task descriptions	Presence of concepts in descriptions
Food & Wine	T1	2,43	3,48	3,81	2,67
	T2	2,33	3,43	3,86	2,43
Travel	T3	2,62	<i>3,57</i>	4,10	2,86
	T4	2,62	<i>3,76</i>	3,95	2,62
Animal	T5	<i>2,76</i>	3,38	3,90	2,67
	T6	2,71	3,14	3,71	2,38
Autos	T7	2,57	2,81	4,05	2,86
	T8	<i>2,86</i>	2,95	3,71	2,71

Note: Lowest values are in **bold**, while highest are in *italics*. Measured with a 5-point Likert scale.

When evaluating the results with respect to different search tasks (i.e. comprehensive, explorative and fact-finding (see Section 3.1)), we observed the biggest improvement were for explorative search tasks, while fact-finding types of task came second. We also observed that the addition of more instances and object properties improved the mean relevance score of fact-finding search tasks, while the addition of sub-classes resulted in the improved performance of exploratory and comprehensive search tasks.

We also observed that it appears that the prototype performs better for shorter queries compared to keyword-based queries. Similar observations for similar approaches have also been observed by Chang et al. (2006). Furthermore, the entity-based queries were generally shorter than the keyword-based queries.

More detail can be found in papers P1 and P6.

4.3.2 Experiment II

Objectives

In this experiment, we focused on the feature vector construction process since the actual search performance depends a lot on the quality of the FVs. In Experiment I (Section 4.3.1), we evaluated the FVs ability to disambiguate search. The evaluation was carried out with real users. When evaluating the results of Experiment I, we found significant dependence between the overall performance and the quality of the ontologies, but we were not able to conclude to what degree the quality of the FVs depends on the quality of the ontologies. Neither, could we conclude how much the quality of the FVs is influenced by the FV construction process and the techniques used. Therefore, the objective of this experiment was to evaluate the sensitivity of the approach with respect to the components of the FVC algorithm and ontologies of different granularity.

Settings

An overview of the feature vector construction (FVC) is depicted in Figure 4.8. The FVC process is composed of three phases. The first phase includes preparing the ontology for further processing. The main aim of second phase was to find candidate

documents that are potentially relevant to the entities. The third phase included grouping documents and identifying the most relevant groups with respect to the ontology. The result of these steps is a list of entities with corresponding FVs that consists of terms associated with both the entities and the domain terminology.

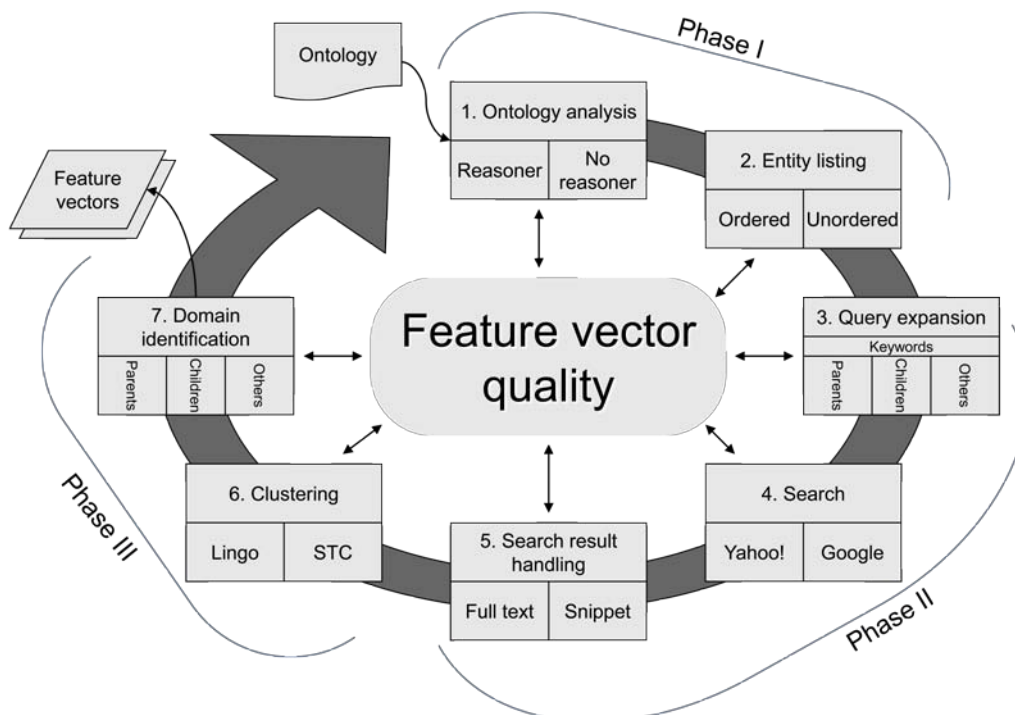


Figure 4.8: An overview of the FV construction process.

To evaluate the sensitivity of the FVC components a set of experiments was performed. For the most extensive experiment, published in paper P4, 29 distinct combinations of FVC parameters were evaluated. To evaluate the sensitivity of the approach, we proposed a set of intrinsic and extrinsic FV quality measures. New measures were proposed since we found that evaluations of semantic search tools are sparse, also acknowledged in a recent paper by Wrigley et al. (2010). Two intrinsic measures were proposed, the Average FV Similarity (AFVS) and the Average FV Neighbourhood Similarity (AFVNS) which both indicate the quality of the FVs with respect to the ontology used. The Average FV NGD⁷ (AFVNGD) is an extrinsic measure that indicates the FV quality with respect to a text corpus (i.e. the Web). Finally, we proposed an overall FV quality score, the FV Quality Score (FVQS), which aggregates the previous three scores. Since the proposed approach is dependent on a backend search engine, we also proposed a method to measure changes provided by the search engine over time (what we have called the Web Drift Effect). These proposed measures are used to assess the quality of the created FVs. Formal definitions are found in paper P4.

⁷ NGD is the abbreviation for the Normalised Google Distance (Cilibrasi & Vitanyi, 2007) that is used in this work to compute the semantic distance between an entity and its FV terms.

Ontologies of different granularity were used to measure their effects on the algorithm. Three ontologies from Experiment I were also used here. The key characteristics of the ontologies are shown in Table 4.5.

Table 4.5: Ontology key characteristics.

<i>Ontology</i>	<i>Classes</i> <i>n/r</i>	<i>Individuals</i> <i>n/r</i>	<i>Properties</i> <i>n/r</i>
Animals	51/51	0/0	0/0
Travel	34/33	14/14	6/6
Wine	82/137	155/194	10/10

Note: n=no reasoner, r=reasoner

Furthermore, the experiments had the following restrictions:

- All OWL object properties were treated as *other* relations.
- Disjointed classes, as a feature, were ignored since siblings were not considered in these experiments.
- The following equality features were ignored: `equivalentClass`, `sameAs`, and `differentFrom`.
- The maximum length of the FVs was set to 30 (top 30 selected by highest frequency) to avoid circumstances with lengthy FVs. However, in Experiment I, no restrictions were put on FV length (the average length was 24).

Results and conclusions

A different set of experiments was conducted. For the most extensive experiment, the process of populating the ontologies took more than 29 hours. The most complex ontology, the Wine ontology, took from 10 to 323 minutes to populate. Furthermore, more than 670.000 queries were processed to assess the quality of the FVs created.

The experiments, described in papers P2-P4, have much of the same settings but were analysed with a different focus in mind. From a component point of view, we concluded the following:

- Step 1 - Ontology analysis:* A reasoner lowers the error rate, but decreases the overall FV quality.
- Step 2 - Entity listing:* Ranking of entities for processing seemed to decrease the FV quality and increase the error rate and is therefore not recommended practice.
- Step 3 - Query expansion:* Query expansion increases the quality of the search results and hence the FV quality. Expanding with parents, children, and other related entities provide the best results in general.
- Step 4 - Search:* Changing between comparable search engines does not seem to yield any major effect if an adequate number of search results are used.
- Step 5 - Search result handling:* Full text documents in combination with the extraction of contextual key-phrases seemed to provide the best positive results but considerably increased the processing time.
- Step 6 - Clustering:* Full text documents provided the best results as in Step 5 (i.e. *Search result handling*). Change of comparable clustering algorithms did not seem to yield any major effect.

Step 7 - Domain identification: Including the parents, children, and other related entities seemed to provide the best results.

With respect to ontologies of different categories, we concluded the following:

Taxonomy type of ontologies (e.g. Animals):

- *Query expansion:* Use of parent entities when expanding the query provided the best results.
- *Clustering input:* Using full text documents in combination with extraction of the most relevant key-phrases seemed to provide the best positive effect on the FV quality.
- *Domain identification:* Including the parents, children, and other related entities seemed to provide the best results when identifying the most prominent cluster candidates.

Lightweight ontologies (e.g. Travel):

- *Query expansion:* Using the parent entities in combination with scope keywords⁸ provided the best results.
- *Clustering input:* Using full text documents in combination with the extraction of the most relevant key-phrases seemed to provide the best positive effect on the FV quality.
- *Domain identification:* Including the parents and other related entities seemed to provide the best results.

More advanced ontologies (e.g. Wine):

- *Query expansion:* Use of parents, children, and other related entities were recommended to provide the best results.
- *Clustering input:* No recommendation. Further research needed, since the Wine ontology used in these experiments is probably not representative.
- *Domain identification:* Including parents and other related entities seemed to provide best results for advanced ontologies.

In general, the most important component with respect to the FV quality is the query expansion component. The parent entities are the most important neighbouring entities. Therefore, FV construction for taxonomy type ontologies (e.g. Animals) is the most sensitive to different techniques, while advanced and rich ontologies, such as Wine, are the least sensitive. This indicates that the FV construction process needs to be tuned; mostly for taxonomy type of ontologies, whereas richer ontologies contain more substance for FV construction and consequently requires less tuning (i.e. the algorithm is not so sensitive to various processing techniques, though the quality can still be improved).

We also found, disconfirming the initial hypothesis, that ranking of entities had a negative effect on the FV quality when compared with the algorithm without ranking.

⁸ Scope keywords are keywords that can represent an ontology as a whole since larger ontologies tend to include several minor domains (e.g. the Wine ontology used also includes an ontology about food). See paper P4 for further information regarding this topic.

Moreover, the total processing time increased (due to the complexity of the ranking algorithm) while the domain identification process took less time (due to fewer comparisons needing to be done).

Furthermore, we found that utilising neighbouring entities when expanding queries yielded better FV quality than using scope keywords⁸. We also found that a high number of search results minimises the difference between the search engines and probably the change in ranking they provide over time.

More details regarding this experiment are found in papers P2-P4.

4.3.3 Experiment III

Objectives

The objective of this experiment was to validate our proposed FV quality measures. In Experiment I (Section 4.3.1), the quality of the FVs was evaluated indirectly, while in Experiment II (Section 4.3.2) we evaluated the quality of the FVs directly by our proposed evaluation measures. Based on the evaluation of the results from Experiment II, we proposed a set of guidelines for the construction of FVs with respect to the components of the algorithms but also ontologies of different categories. The quality measures provided a mean to estimate the output of different configurations of the algorithm. However, this needed to be related to the actual performance in a search application.

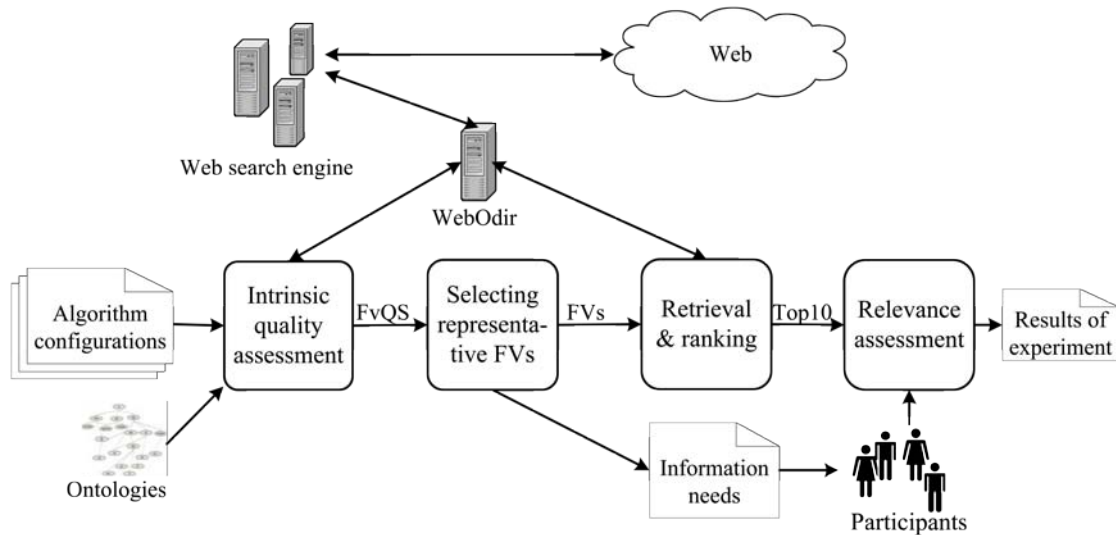


Figure 4.9: Design of Experiment III.

Settings

The design of the experiment is depicted in Figure 4.9. Since we wanted to validate the evaluation measures proposed in Experiment II, the same set of ontologies (see Section 4.3.2) was also used in this experiment. The FVs were constructed using different algorithm configurations. The quality of the FVs created was assessed using the same measures as proposed in Experiment II. A subset of these FVs (i.e. entities) was selected using a set of criteria, and finally evaluated by the test subjects. First, for each of the

selected entities we formulated a query that was based on the entity labels. In addition, a set of corresponding simulated information needs was formulated. Then, the top ten results for each query were recorded and evaluated by the test subjects. The search results were evaluated with respect to the simulated information needs.

The test subjects were presented with three simulated information needs with corresponding search results retrieved from our semantic search system. Each query was submitted three times using distinct FVs. The FVs were created with dissimilar quality parameters, i.e. low, medium, and high (a summary of the different parameters are depicted in Table 4.6). As a result, the semantic search system retrieved three different search results (i.e. low, med, and high) for each of the selected entities. In total nine queries were submitted and nine search results were recorded. For each search result, the 10 top ranked documents were selected for evaluation. Consequently, each user needed to assess 90 retrieved documents.

Table 4.6: Summary of quality parameters used to construct the FVs.

	<i>Low Quality Parameters</i>	<i>Medium Quality Parameters</i>	<i>High Quality Parameters</i>
<i>Ontology analysis</i>			
With reasoner			X
Without reasoner	X	X	
<i>Query expansion</i>			
Parents		X	X
Children			X
Others			X
<i>Search results</i>			
Number of results	100	100	100
<i>Domain identification</i>			
Parents	X	X	X
Children		X	X
Others			X

Results and conclusions

Nine subjects took part in the experiment, which were mainly colleagues at the Norwegian University of Science and Technology (NTNU). They were not offered any form of compensation for their used time; instead, an amount of money was donated to the Red Cross, an international humanitarian relief agency, on behalf of each of the participants.

Populating and analysing the ontologies took more than 10 hours; the most complex ontology, the Wine ontology, took between 133 to 197 minutes to both populate and analyse. When populating the ontologies and evaluating the quality of the FVs, more than 20.000 queries were submitted to Google[®].

Each of the ontologies was populated with three different quality construction parameters as summarised in Table 4.6. Table 4.7 summarises the results of these quality parameters for the selected entities. The quality of the FVs was assessed with the proposed quality measures.

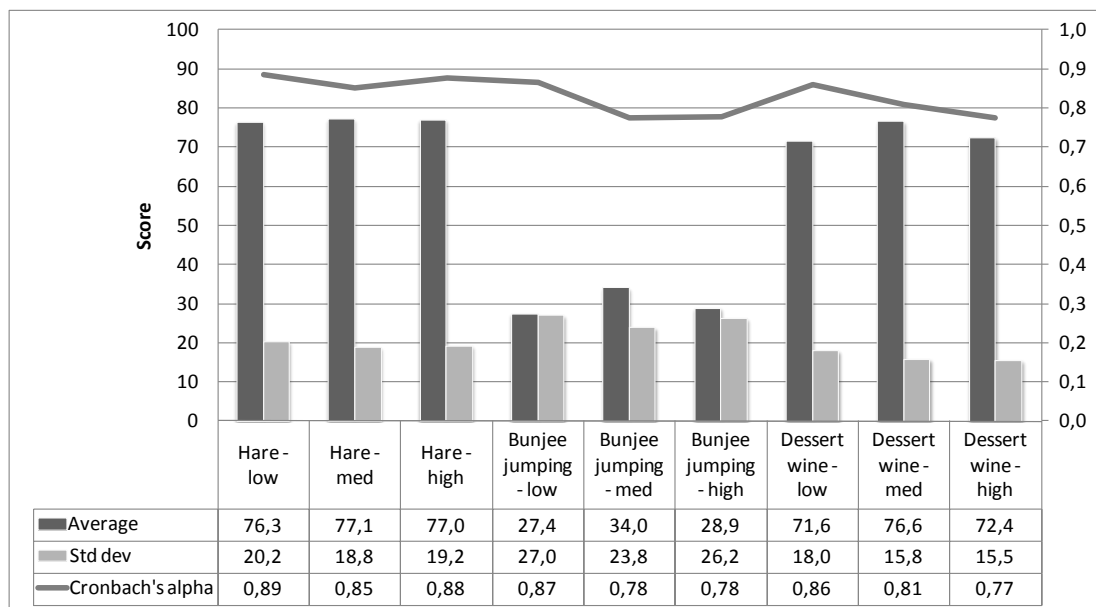
The test subjects assessed the relevance of the search results with respect to the simulated information needs. Each information need was designed to reflect the corresponding generic query formulated for each entity selected. Figure 4.10 provides an overview of the search result relevance scores (i.e. the average of the scores) along with the standard deviation and the Cronbach's alpha scores.

Table 4.7: FV quality scores w.r.t. different construction parameters.

Ontology	Entity	FvNS			FvNGD			FvQS		
		<i>Low</i>	<i>Med</i>	High	<i>Low</i>	<i>Med</i>	High	<i>Low</i>	<i>Med</i>	High
Animals	Hare	<i>0,251</i>	0,609	0,911	<i>0,283</i>	0,244	0,240	<i>0,670</i>	0,741	0,775
Travel	Bunjee Jumping	<i>0,014</i>	0,200	0,570	<i>0,130</i>	0,107	0,105	<i>0,784</i>	0,824	0,862
Wine	Dessert Wine	<i>0,642</i>	0,763	0,911	<i>0,158</i>	0,151	0,130	<i>0,822</i>	0,841	0,874

Note: The best values are highlighted in **bold**, while the lowest values are highlighted in *italics*.

We observed in Figure 4.10 that the relevance scores are high except for the *Bunjee jumping* entity, which can be explained by the fact that the entity is misspelled and consequently results in less relevant search results. Furthermore, we observed that the search results, where assumed medium quality parameters were used, provided the best results, while the low quality parameters provided the lowest scores.



Note: The relevance scores are in the range [-50, 100].

Figure 4.10: Relevance scores and Cronbach's alpha for selected entities.

We analysed how well these results matched the assessed quality of the FVs using the proposed evaluation measures (the results are summarised in Figure 4.11). We found the FVs created with medium and high quality parameters provided better results than the FVs created with low quality parameters when evaluated by both the test subjects and the proposed quality metrics (see the scores for *Med* and *High* vs. *Low* in Figure 4.10 and *Med* and *High* in Figure 4.11). However, the participants scored the search results where the medium quality FVs were used the highest. Therefore, we concluded that our proposed overall FV quality score, being an aggregated score of the other proposed scores, needs to be revised to better reflect the FVs used in the search.

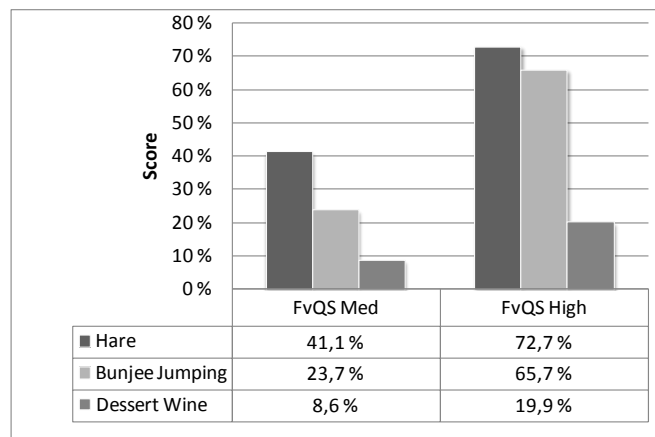


Figure 4.11: Top 1 FV quality scores relative to the lowest score.

More details can be found in paper P5.

4.4 Synopsis of main publications

In this section, we present the published results as part of this work. For each paper, a structured abstract is provided along with remarks about its relevance to the thesis and the contributions made by the authors.

4.4.1 Feature vector construction

P1: Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect

Brief summary of contents

In this paper, we tackle the particular problem of heterogeneity in search resulting from discipline specific languages used in documents. We propose an approach for the construction of semantic-linguistic feature vectors (FV). These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from documents on the Web. We explain how these FVs are constructed and provide an example of the construction process. Furthermore, we present a conducted experiment and discuss how the use of these FVs influences the search performance.

Objectives

The objective of this paper is to describe and evaluate our unsupervised approach to feature vector construction (FVC).

Contributions

The contributions of this paper are the proposed formal definition of a feature vector, the proposed FV construction algorithm, and the evaluation of FVs used to disambiguate search. We evaluated the approach by the means of a controlled experiment and a post-task questionnaire.

Results, observations and conclusions

A feature vector constitutes a rich representation of an entity that is related to the actual terminology used in a text corpus. It is represented as a two-tuple with a semantic and a linguistic enrichment part. The semantic enrichment part represents a set of neighbourhood entities and properties in an ontology, while the linguistic part is an enrichment of an entity with a set of terms in significant proximity to the entity and its semantic neighbourhood. A more formal definition of a feature vector is found in the paper. The algorithm to construct such feature vectors is presented in Section 4.2.2 and illustrated in Figure 4.5, while more details about the algorithm are found in the paper.

Table 4.3 summarises the main results of the experiment. The four ontologies used in this experiment were modified by adding instances (all ontologies), specifying additional object properties (Travel, Animal and Wine ontologies) and introducing equivalent classes (Animal and Auto ontologies). These modified ontologies were denoted as *ontology ver. 2*. The differences in granularity affected the quality of the FVs. How these affect the search performance can be observed in Table 4.3, where we see that the modified ontologies (i.e. ontology ver. 2 vs. ver. 1) yielded an improvement in the mean score by 10.7%. This indicates that in general a more advanced ontology, in the sense of having more relations, properties, and individuals, does perform better than a similar simpler ontology. A reason for this is that the more advanced ontologies contain more information in a form of rich relationships and instances that directly contribute to better quality of the entity FVs and hence will contain less noise compared to those of a simpler ontology. One of the reasons for the modified version of the Travel ontology performing worse than the original version is the quality of those ontologies (Strasunskas & Tomassen, 2008a).

We experimented with an intrinsic and extrinsic evaluation of the quality of the FVs. The intrinsic measure included calculating the average similarity between all the FVs of an ontology. This score gave us an indication of the uniqueness of the FVs, where zero indicates that all the FVs are unique while one shows that all are equal. In this experiment, we were not able to find a direct correlation between this uniqueness score and the user obtained relevance score. To get an extrinsic indication of the FV quality we experimented with the Normalized Google Distance (NGD) measure, introduced by (Cilibrasi & Vitanyi, 2007). NGD utilises the number of hits by Google (or any other search engine) for two selected keywords and the combination of those to calculate a semantic distance between them. We found indications that in general version 2 of the ontologies (except for the Wine ontology) have a higher semantic similarity than version 1. The differences between ontology versions 1 and 2 for Animals, Autos, Travel, and Wine are 2,04%, 6,16%, 4,73% and -0,87%, respectively. Another interesting observation is that the Animals and Autos ontologies have a lower semantic similarity than Travel and Wine, which can be explained by the better quality of latter ontologies (Strasunskas & Tomassen, 2008a).

In this paper, we presented a prototype that was developed, and real users evaluated its performance. An experiment was conducted and the analysis of the experiment showed that the approach performed well in some domains but worse in others (see Section 4.3.1). We also showed that adding more instances and specifying additional object properties to the ontologies in general positively affected the quality of the FVs and

hence improved the search performance. However, some changes that were made had a negative effect on the quality. Therefore as one of the future tasks, we concluded that we needed to categorise the ontologies according to different key characteristics to find trends relevant to these categories. Moreover, the experiment showed the need to look into alternative techniques in order to reduce the sensitivity of the approach with respect to the quality of ontology.

Relevance to the thesis

This paper introduced the FVC algorithm used in Prototype I, presented in Section 4.2.2, and its evaluation with real users (i.e. Experiment I presented in Section 4.3.1). The quality of the FVs was indirectly evaluated by how they performed in disambiguate search.

Author contribution

This paper was mainly written by Tomassen. Strasunskas helped with the analysis of the results and contributed with comments and refinements to the paper.

P2: Semantic-Linguistic Feature Vectors for Search: Unsupervised Construction and Experimental Validation

Brief summary of contents

In this paper, we elaborate on an approach to the construction of semantic-linguistic feature vectors that are used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. The value of this approach is twofold. We focus on aspects of the components of the FV construction algorithm and their affect on the feature vector quality. We analyse the effect of alternative techniques and lay down recommendations and lessons learned.

Objectives

The objective of this paper is to describe and evaluate our unsupervised approach to feature vector construction (FVC). In this paper, we go deeper into the FVC algorithm and focus on its components and their affect on the FV quality.

Contributions

The contributions of this paper are the FV construction algorithm and the evaluation of the quality of constructed FVs. We evaluated the FV construction algorithm by means of a controlled experiment.

Results, observations and conclusions

In this paper, both intrinsic and extrinsic FV quality measures were proposed that were based on contemporary literature and lessons learned (mainly from paper P1). In total four measures were proposed (listed in Section 4.3.2). Formal definitions of the proposed evaluation measures are found in the paper.

Twenty-three tests were conducted and the results analysed. We found the affect of each of the components and draw some general conclusions that were independent of

ontology quality. We found that query expansion increased the quality of the search results and hence the quality of the FVs. Furthermore, we found that inclusion of parents, children, and other related entities provided the best results. For the search results and clustering components, we found full text documents in combination with the extraction of the most relevant key-phrases provided the best positive effect on the FV quality. However, download of each page probably increased the processing time considerably compared to using just snippets. To identify the most prominent cluster candidate we found that including the parents, children, and other related entities provided the best results.

We also found the query expansion component to be the most important component with respect to the FV quality. Furthermore, the parent entities were the most important neighbouring entities when both expanding a query and identifying the most prominent candidate cluster. In addition, we found that a high number of search results minimised the difference between the underlying search engines and probably the change in ranking they provided over time.

Moreover, we did a few tests with the NGD measure to assess the semantic distance between the entities within the ontologies. Preliminary results indicated that there is a connection between the findings and characteristics of each ontology and the assessed NGD ontology score. However, because of the scale of these tests it needs to be explored further.

Relevance to the thesis

In paper P1, we introduced the FVC algorithm used in Prototype I, presented in Section 4.2.2, and the quality of the FVs was indirectly evaluated by how they performed disambiguate search. In this paper, we went deeper into the FVC algorithm used in Prototype I and conducted a new evaluation (i.e. Experiment II presented in Section 4.3.2) than the one presented in paper P1.

Author contribution

This paper was mainly written by Tomassen. Strasunskas helped with the analysis of the results and contributed with comments and refinements to the paper.

P3: Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation

Brief summary of contents

In this paper, we elaborate on a new algorithm used to construct semantic-linguistic feature vectors (FV) that are used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. We focus on this new feature vector construction (FVC) algorithm and evaluate the FV quality with respect to a set of heterogeneous ontologies.

Objectives

The objective of this paper is to present and evaluate the proposed, and new, FVC algorithm. We evaluated the FV construction algorithm by means of a controlled experiment.

Contributions

The contribution of this paper is the new FVC algorithm and the evaluation of the constructed FVs. The quality of the FVs were evaluated by means of a controlled experiment.

Results, observations and conclusions

In this paper, a new FVC algorithm was presented. The biggest differences between this algorithm and the one proposed earlier are how the entities are prepared and later used to identify the most prominent cluster of candidate terms for each entity. The hypothesis is that more information is available for the most central entities and therefore provide better candidates to discriminate relevant clusters of candidate terms. Another assumed upside effect is less similarity calculations and hence the algorithm becomes more efficient. The algorithm is described in Section 4.2.3 and illustrated in Figure 4.6, while more details about this algorithm are found in the paper.

To evaluate the effect of this new algorithm, we used the same measures introduced in paper P2 to evaluate the FV quality. A set of 10 experiments were conducted on three ontologies resulting in a total of 30 different configurations (the general settings of this laboratory experiment are described in Section 4.3.2). The process of constructing the FVs took more than 13 hours in total; the most complex ontology, the Wine ontology, took from 16 to 298 minutes to process. When evaluating the quality of the FVs using the NGD measure, more than 260.000 queries were submitted.

Based on the analysed results, we found that the ranking of entities had a negative effect on the FV quality when compared with the previously proposed algorithm that did not use ranking. Moreover, surprisingly the total processing time increased, mainly because of the complexity of the ranking algorithm. The domain identification process, on the other hand, took less time because fewer similarity calculations were needed.

We also analysed the results of the algorithms with respect to a set of heterogeneous ontologies and came up with a set of recommendations (these are listed in the paper and in Section 4.3.2). We found taxonomy like ontologies (e.g. Animals) to be the most sensitive to the different techniques used while advanced or rich ontologies (e.g. Wine) to be the least sensitive. This indicates that the FVC process needs to be tuned mostly for taxonomy type of ontologies, whereas richer ontologies possess more knowledge and hence are less sensitive to parametric changes. The knowledge contained in the ontologies provided a good enough basis for FVC (i.e. the construction process was not so sensitive to the processing techniques), though the quality of the FV could still be improved.

Relevance to the thesis

The contribution of this paper is the new FVC algorithm (i.e. that used in Prototype II and presented in Section 4.2.3) and its evaluation (i.e. Experiment II presented in Section 4.3.2). In paper P1, the FVC algorithm of Prototype I (Section 4.2.2) was described as consisting of two phases, while the algorithm presented in this paper has been extended to three phases.

Author contribution

This paper was mainly written by Tomassen. Strasunskas helped with the analysis of the results and contributed with comments and refinements to the paper.

4.4.2 Feature vector quality

P4: Measuring intrinsic quality of semantic search based on Feature Vectors

Brief summary of contents

In this paper, we analyse a process of constructing semantic-linguistic Feature Vectors (FV) used in our semantic search approach. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. We focus on the process of FV construction and the impact of chosen techniques on the quality of FVs. We report on a set of laboratory experiments and analyse aspects affecting the FV quality and the FV construction error rates. In previous papers, we have proposed a set of intrinsic and extrinsic measures to assess the quality of feature vectors. In this paper, two additional measures are proposed as being relevant when FVs are evaluated using a constantly changing text corpora (e.g. uncontrollable external changes like change of ranking by the search engine provider). Since the Web was used to evaluate our FVs, we proposed to measure both short- and long-term changes of the Web, what we have called the Web Drift Effect. Furthermore, we investigate the effect of the ontology's granularity on FV quality.

Objectives

The objective of this paper is to present our proposed method for evaluation of feature vector quality with respect to both the ontology and text corpus used.

Contributions

The contribution of this paper is as a method to evaluate the quality of feature vectors, Web drift effect measures and lessons learned. We evaluated the approach by means of a controlled experiment.

Results, observations and conclusions

In total, 32 experiments were conducted (introduced in Section 4.3.2, while more details are found in the paper). A set of ontologies were populated, which took over 29 hours.

To measure the Web Drift Effect, a set of ontologies were populated with basic parameters both in the beginning and at the end of the experiment. Then we measured

the change in FV quality. The short-term effect is relevant to changes than can occur during, e.g. an experiment (the experiment presented in this paper was conducted over a week) while the long-term effect is relevant for measuring changes that have occurred since, e.g. the last performed experiment (in this experiment we measure the changes from our previously conducted experiment). In this experiment, the short-term effect was used as the standard deviation for the measurements. When measuring the long-term effect, we found the quality of the FVs increased by 2,7% for the Animals ontology, while there was a slight decrease for the Travel and Wine ontology when we compared them with the baseline. We observed the same trend for the baseline. However, the differences were less than expected when compared to the baseline.

Based on an analysis of the results we found aspects with each component of the FV construction process that affected the quality of the FVs (these are listed in the paper and in Section 4.3.2). We found the *query expansion* component (component 3 in Figure 4.8) affected the FV quality most. Parent entities are, in general, the most important neighbouring entities for both query expansion and domain identification (component 7 in Figure 4.8). Furthermore, expanding queries with neighbouring entities yielded better FV quality than expanding them with scope keywords. We also found that a high number of search results minimised the difference of alternative search engines.

Relevance to the thesis

In this paper, we evaluated both our FVC algorithms introduced in Prototypes I and II (Sections 4.2.2 and 4.2.3 respectively). We focused on aspects of the components of the feature vector construction algorithms (introduced in papers P1 and P3) that affect the FV quality. In the evaluation, we analysed the effect of alternative FV construction techniques on the quality of the FVs. We also tried to predict the potential search improvements based on the findings from these experiments and the experiences from the work published in paper P6.

Author contribution

This paper was mainly written by Tomassen. Strasunskas helped with the analysis of the results and contributed with comments, designing the experiment, writing and refining the paper.

P5: Constructing Feature Vectors for search: investigating intrinsic quality impact on search performance

Brief summary of contents

In this paper, we revisit our approach to construction of semantic-linguistic feature vectors that are used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. We have proposed a method for the evaluation of feature vector quality. The quality of the FVs are measured with respect to both the ontology and text corpus used. In this paper, we validate the proposed evaluation method with respect to their ability to disambiguate search. We can conclude that the proposed metrics provide good indications of the

quality of the FVs. Nevertheless, the results also suggest that the proposed metrics need to be revised to fit the needs of search applications.

Objectives

The objective of this paper is to validate our proposed method for evaluation of FV quality. More specifically, relate the FV quality (by defined metrics) to actual performance (assessed by end-users) in search.

Contributions

The contribution of this paper is the validation of our proposed method for evaluation of feature vector quality. The proposed evaluation method was validated in a controlled experiment with real end-users.

Results, observations and conclusions

Real users, mainly colleagues from the university, evaluated the search performance of our application. A set of ontologies were populated using different quality settings (low, medium and high quality parameters). Then, we assessed the quality of the FVs using our proposed evaluation measures. Next, a set of queries was created based on the selected entities (i.e. the entity labels) to be evaluated that satisfied defined criteria. The queries were submitted to the underlying search engine and corresponding FVs were used to filter and re-rank the results. The top-ten results for each query were evaluated by the test subjects. The result sets were evaluated with respect to a set of simulated information needs. Each test subject evaluated 90 retrieved documents. More information about the experiment settings can be found in Section 4.3.3 and the paper.

When evaluating the results of the experiments we found ontologies populated with medium FV quality parameters provided the best search result scores while those populated with the lowest FV quality parameters provided the lowest scores. However, the medium FV quality parameters scored marginally better than the high quality parameters. Nevertheless, the findings indicated that our proposed metrics provide a good indication of the quality of the FVs, while the aggregated overall FV quality score needs to be revised. Since the overall FV quality score is an aggregated linear score it cannot be tuned to fit the observations done in this experiment, but needs to be revised to better reflect the FVs used in search. More details are in Section 4.3.3 and the paper.

Relevance to the thesis

In paper P4, we proposed an evaluation method based on analysis of components' sensitivity with regards to the quality of the resulting FVs. The proposed metrics were analytically derived from contemporary literature. In this paper, we relate the measured quality of the FVs with their actual performance in a search application. We investigated the performance of the overall approach related to different qualities of FVs.

Author contribution

This paper was mainly written by Tomassen. Strasunskas helped with the analysis of the results and contributed with comments and refinements to the paper. The design of the experiment was a joint effort.

4.4.3 Feature vector applications

P6: An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks

Brief summary of contents

In this paper, we present our approach to semantic search where entities in an ontology are associated with domain terminology by feature vectors. A FV reflects the semantic and linguistic neighbourhoods of a particular entity. The semantic neighbourhood is derived from an ontology and is based on related entities and specified properties, while the linguistic neighbourhood is based on co-location of terms in a text corpus. The FVs are created offline and later used online to filter and re-rank the results from an underlying search engine. We elaborate on the approach and describe how FVs are constructed. Then we report on an experiment where we analyse the sensitivity of the approach with respect to ontology quality and search tasks.

Objectives

The objective of this paper was to present the proposed approach and analyse both how ontologies of different granularity and search tasks affect search performance.

Contributions

The contributions of this paper lie in the initial analysed results of how ontologies of different granularity and search tasks can affect the overall performance of semantic search. We evaluated the approach by means of a controlled experiment and a post-task questionnaire.

Results, observations and conclusions

Results indicated that the proposed approach and prototype implemented are able to improve the search performance of a standard Web search engine by more than 10% on average. Furthermore, the analysis of the experiment data shows that the level of ontology specification was important for the quality of the FVs. Recall from the experiment settings (described in Section 4.3.1) that ontology version 2 is an altered edition of version 1 with different granularities and levels of knowledge specification. More advanced ontologies, in the sense of having more relations, properties and individuals, perform better than similar simpler ontologies. Analysis of the results showed an improvement in performance of 10,5% due to the enhanced quality of the ontologies.

However, when evaluating the results, we found that some topics (i.e. simulated information needs, see Appendix E) performed better than others. For example, topic 6 (see Table 4.4) had the lowest relevance score but also scored lowest on the task

description and on the presence of concepts in the description. Consequently, it was more difficult to formulate useful queries. The topic familiarity and ontology usefulness also received third lowest rates. Furthermore, topic 6 had the biggest variance in query length. All these factors surely contributed to the low relevance score for this topic.

The analysis of the results also showed that users tended to formulate shorter queries for the entity-based approach versus the traditional keyword-based approach. Recall from the experiment settings (described in Section 4.3.1) that the participants were divided into two sub-groups. The first group were required to formulate the keyword-based queries prior to the entity-based queries and the other sub-group vice versa. The first group formulating the entity-based queries used in average 13% fewer keywords and 14% fewer entities compared to the second group formulating the entity-based queries last. While the second group when formulating the keyword-based queries first had a tendency to use most of the keywords in the entity-based search as well, consequently producing longer entity-based queries than the first group. The keyword-based queries were almost equal in length with a difference of only 2% between both groups. These observations indicate that the participants have a prior expectation of such systems and hence apply the learnt way of search on this new search system.

For shorter entity-based queries, the prototype seemed to perform better when compared to similar keyword-based queries. Similar findings are also observed for other entity-based approaches (e.g. (Chang et al., 2006)). In addition, analysis of the results from the questionnaire showed that the participants found the proposed approach particularly helpful in formulating queries for unfamiliar domains.

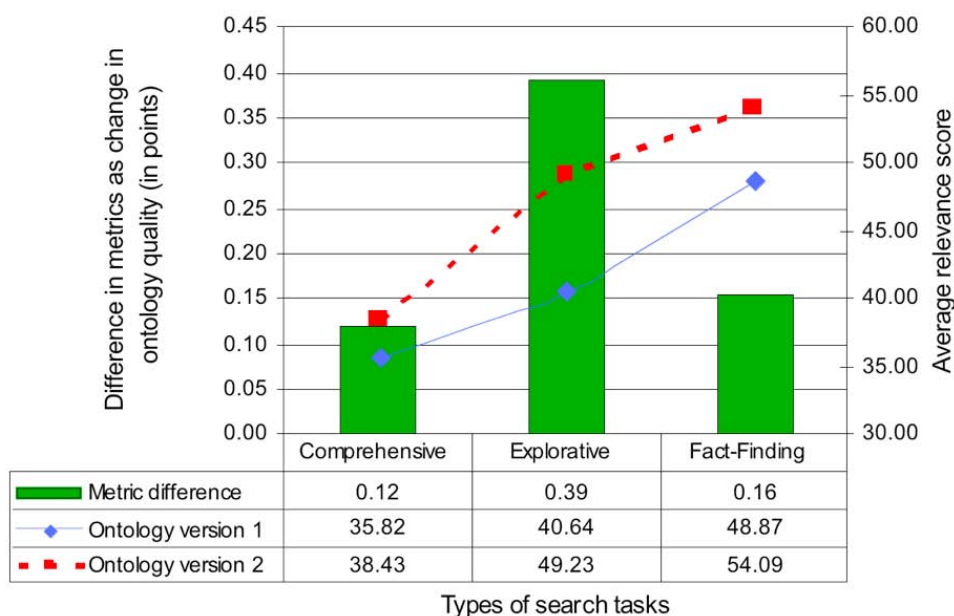


Figure 4.12: Comparison of ontology quality and search performance w.r.t. search tasks.

We also analysed how the approach performed in various search tasks. The analysis showed that certain ontology elements have bigger effect on certain information tasks than other ontology elements (see Figure 4.12). Furthermore, the approach exhibited the best performance for fact-finding kinds of search task, producing an almost 50% higher relevance score when compared to the comprehensive kind of search task.

Relevance to the thesis

In paper P1, we focused on the FVC algorithm used in Prototype I, presented in Section 4.2.2, and the quality of the FVs was evaluated by how they performed disambiguate search. In this paper, we are evaluating the same prototype (i.e. Prototype I) but with a focus on the semantic search capabilities of the prototype. We analyse how the approach performed with respect to ontologies of different granularity and three different search tasks. Furthermore, we analysed the results from the questionnaire.

Author contribution

This paper was mainly written by Tomassen. Strasunskas contributed with the search strategies and the EvOQS framework. In addition, he helped with the analysis of results and contributed with comments and refinements to the paper. Design of the questionnaire and experiment was a joint effort.

P7: Cross-Lingual Information Retrieval by Feature Vectors

Brief summary of contents

In this paper, we propose a novel approach to cross-lingual information retrieval (CLIR) based on feature vectors (FV). We investigated query translation in CLIR, especially the challenges caused by ambiguity and homonym. We based our ideas on FVs and our approach used the context of the queries during the translation. Achieving good query translation can be difficult, due to short queries lacking context information. Using information external to the query (i.e. FVs that are based upon ontologies and documents) can reduce the effect of ambiguity and homonym in queries. Therefore, we argue that direct translation of FVs can be sufficient for information retrieval applications. Different approaches for translation of these FVs are proposed and discussed.

Objectives

The objective of this paper is to present and discuss our proposed approach to cross-lingual information retrieval (CLIR) based on FVs. We present and discuss two different approaches to query translation.

Contributions

The contribution of this paper is our novel approach to CLIR based on FVs. However, we have not fully implemented and tested the approach. Consequently, we cannot conclude the success of this approach before it has been verified in a full-scale test.

Results, observations and conclusions

In this paper, we presented two different approaches for how FVs can be translated. The quality of the translations were measured by carrying out a double translation (i.e. translation into a chosen language and then back again into the original language). Obviously, the best result is provided if a twice-translated FV becomes equal to the original FV.

In the first approach, direct translation was used where each term was translated independently of each other. To automatically find the correct translation of a term is typically very difficult because terms are ambiguous and consequently can have many different meanings dependent on their context. For the first approach the context was disregarded, therefore we selected the first translation proposed by the dictionary used. Not surprisingly, this approach produced poorly translated FVs.

In the second approach, the semantic relations between the terms were used in the translation process. Since the context was considered during the translation process the translated FVs were of much higher quality than the first approach. In fact, in our exemplified results, the twice-translation gave a 100% match with respect to the original FV. However, in this paper we presented a small experiment with only one example. Consequently, more thoroughly testing needs to be done before we can conclude how successful this proposed approach is in general.

Relevance to the thesis

In paper P6, we showed how our approach to semantic search using FVs can be used to disambiguate search queries, while in paper P8 we showed how the approach can be used to support scenario-driven information retrieval. To show the applicability of our approach in this paper we proposed the use of FVs to support cross-lingual information retrieval.

Author contribution

Lilleng and Tomassen wrote this paper. Lilleng focused on the multilingual parts while Tomassen focused on the system parts of the paper. Both contributed with comments and refinements to the paper.

P8: Scenario-Driven Information Retrieval: Supporting Rule-Based Monitoring of Subsea Operations

Brief summary of contents

Production systems used by the subsea petroleum industry are knowledge and information intensive. Any problem needs to be solved quickly and efficiently avoiding decommissioning or waiting for the symptoms to be escalated. This requires precise information to be supplied on time. For this reason, we have proposed rule-based monitoring of device performance. However, covering all possible cases by rules is labour-intensive and not a trivial task. Therefore, we propose a scenario-driven information retrieval (IR) approach to complement rule-based monitoring. We elaborate on the proposed approach and how it can be integrated in rule-based systems in order to support incomplete inference, employ scalability and efficiency of IR engines. The main objective is to automatically formulate a query that is sent to an IR engine every time an incomplete inference happens (i.e. when a specific case has no rules defined).

Objectives

The objective of this paper is to elaborate on task-specific information retrieval and how it can be integrated in rule-based systems in order to support incomplete inference, employing scalability and efficiency of retrieval engines.

Contributions

In this paper, we proposed the use of FVs to support scenario-driven information retrieval. Rule-based approaches can be applied to condition monitoring of subsea production. However, not all possible cases can be encoded in rules beforehand. Therefore, the contribution of this paper is our proposed approach to complement rule-based monitoring with task-specific and ontology-based information retrieval where FVs are used. However, we have not fully implemented and tested the approach and cannot conclude on the success of this approach before it has been fully tested.

Results, observations and conclusions

In order to complement rule-based monitoring, we proposed an approach to scenario-driven information retrieval that is evoked every time incomplete inference happens. We adapted our semantic search system to support rule-based processes of production monitoring (i.e. integrated structured data and knowledge with unstructured information provided in natural language documents). The entities in an ontology are associated with contextual task terminology in terms of FVs, thus tailoring the ontology to the content of the text corpus. This adaptation is fundamental in order to provide useful and usable services to a variety of users in the presence of large variations in resources and activities. The task-specific FVs are later used to enrich a provided query and hence provide means to bridge the gap between the query terms and the terminology used in textual documents.

The research reported in this paper was not fully implemented. Consequently, we were not able to fully evaluate this approach and hence could draw conclusions on the feasibility of this approach. Nevertheless, smaller experiments showed promising results.

Relevance to the thesis

In paper P6, we showed how our approach to semantic search can be used to disambiguate search queries. To show the applicability of our approach we proposed in paper P7 to use FVs to support cross-lingual information retrieval. In this paper, we showed further applicability of our approach in how it can be used to support scenario-driven information retrieval.

Author contribution

This paper was mainly written by Strasunskas. Tomassen mainly focused on the use of FVs in search and contributed with comments and refinements to the paper.

5 Evaluation

In this chapter, we evaluate the results presented in Chapter 4. First, in Section 5.1, the research questions (Section 1.4) are evaluated with respect to the published papers (Part II). In Section 5.2 we evaluate the contributions (Section 1.6) with respect to both the research questions and the published papers. In Section 5.3 we evaluate the contributions with regards to related work. Then we look upon the relevance of the contributions in Section 5.4 and their validity in Section 5.5.

5.1 Research questions revisited

In this section, we revisit the research questions (introduced in Section 1.4) and evaluate whether they have been answered in the published papers (Part II). Table 5.1 presents an overview of the research questions and how they relate to the published papers provided in Part II.

Table 5.1: Published papers answering research questions.

Papers	Research questions			
	<i>RQ1</i>	<i>RQ2</i>	<i>RQ3</i>	<i>RQ4</i>
<i>P1</i>		X		
<i>P2</i>		X	X	X
<i>P3</i>		X	X	X
<i>P4</i>			X	X
<i>P5</i>			X	X
<i>P6</i>	X	X		
<i>P7</i>	X			
<i>P8</i>	X			

RQ1: Can the retrieval effectiveness of search systems be improved by utilising ontologies?

Answering RQ1 was the focus of the *Analysis and design* phase (Section 2.3.1). A broad literature study was conducted that led to our proposed approach based on the concept of FVs. The proposed approach was introduced in papers P6-P8. In papers P7 and P8 we proposed two potential search applications using FVs. In paper P6 we proposed our semantic search approach that was evaluated by real users. In paper P7 we proposed FVs used to support cross-lingual IR, while in paper P8 we proposed the use of FVs to support scenario-driven IR. The proposed applications published in P7 and P8 were based on preliminary prototypes, while the approach described in P6 was fully implemented (Prototype I, described in Section 4.2.2).

Prototype I was evaluated with real users. We found FVs to be an effective approach to connect terminologies provided in ontologies and textual documents. For example, the evaluation of Experiment I (Section 4.3.1 and paper P6) revealed that the prototype

improved the retrieval effectiveness⁹, on average by more than 10% compared to the baseline. The results showed that the approach is capable of handling ambiguity in search queries. Furthermore, we found the relevance scores for fact-finding types of search task to be almost 50% higher if compared to comprehensive types of search task. We also observed that the users tended to formulate shorter queries for entity-based queries than for keyword-based queries. In addition, shorter entity-based queries seemed to perform better when compared to similar keyword-based queries.

RQ2: How can the terminology provided in an ontology be related to terms in textual documents and queries?

The ultimate goal of the literature study conducted in the *Analysis and design* phase (Section 2.3.1) was to find an approach to connect the terminology provided in ontologies with corresponding terminology found in textual documents. In paper P1 and P6, an approach based on semantic-linguistic FVs was proposed, implemented and evaluated. In paper P1, we described in detail how domain terminology encoded in ontologies can be connected to the actual terminology used in textual documents by means of FVs. An advantage of the proposed approach is that it is independent from a set of relevant documents, i.e. a diverse corpus like the Web can be used. In paper P2 the quality of these FVs was evaluated more thoroughly, while in paper P3 an alternative FV construction algorithm was proposed. In paper P6 we showed how FVs can be used to extend an existing search system with semantic capabilities. The underlying search system does not have to be aware of this extension, that is no changes are made to the core components (i.e. indexing and ranking) of the system only the component handling the presentation of the results.

RQ3: How can the quality of the associations between the concepts of an ontology and a text corpus be evaluated?

Based on the studied literature we found that approaches for the evaluation of the quality of FVs (or similar approaches) was scarce. The approaches were mostly evaluated indirectly, i.e. based on their performance of a designated application. Consequently, it became difficult to both compare these approaches and assess whether they are optimal. Therefore, in the papers P2-P5 we proposed a set of intrinsic (AFVS and AFVNS, described in the papers) and extrinsic (AFVNGD, also described in the papers) FV quality measures. In addition, we proposed an overall FV quality score (FVQS) as an aggregated score of the three former scores. In general, we found the query expansion component to be the most important with respect to the FV quality.

In paper P5, we validated the proposed scores using real users and found them to provide good indications of the quality of the FVs. However, we also found that the overall aggregated score (i.e. FVQS) needs to be revised to better fit the needs of a

⁹ In this experiment, the retrieval effectiveness (see Section 1.2) was measured by assessing the relevance of the 10 top ranked documents retrieved by the system for each submitted query (formulated by the users). The relevance of the documents was assessed based on the participants' perception of the topic descriptions. In addition, the participants completed a questionnaire of 29 questions. However, the questionnaire evaluation results were not included in the average retrieval effectiveness score.

semantic search. The quality was assessed with respect to the ontologies, the text corpus (i.e. the Web) and the construction process.

In paper P3 we also wanted to explore whether the approach is independent of the ontology entities processing sequence. We concluded that ranking of entities had a negative effect on the FV quality when compared with the algorithm without ranking. Note that the ranking of the entities is not the issue but rather how it is used in the processing of the entities by the algorithms (details about the differences of these algorithms are found in Section 4.2).

RQ4: What features of an ontology influence the search performance?

Answering RQ4 was part of the focus of the third research phase (Section 2.3.1) of this work. In papers P2-P5 we focused on the feature vector construction process, in particular aspects of the components and the effect of applying different construction techniques. We also evaluated the sensitivity of the FVC algorithms with respect to a set of heterogeneous ontologies of different granularity (i.e. categorised as taxonomy, lightweight and advanced kinds of ontology). The results were analysed and we found trends with respect to these categories, indicating that different ontology categories need construction parameters tuned for each category. Furthermore, we found taxonomy type of ontologies being the most sensitive to different techniques, while advanced and rich type of ontologies being the least sensitive. Based on these findings a set of guidelines were proposed (these are listed in Section 4.3.2 and in the papers). Furthermore, in paper P5 we validated these findings by evaluating the approach using real users.

5.2 Contributions

In this section, we evaluate the contributions (Section 1.6) with respect to the published papers (Part II). Table 5.2 summarises the relationships between the contributions and the published papers.

Table 5.2: Relationships between the contributions and the published papers.

Contributions	Papers							
	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>	<i>P6</i>	<i>P7</i>	<i>P8</i>
<i>C1</i>	(x)					X		
<i>C2</i>						(x)	X	X
<i>C3</i>	X	(x)	(x)	(x)				
<i>C4</i>		X	X	(x)				
<i>C5</i>		(x)	(x)	X	X			

Note: Minor contributions are denoted by (x).

C1: An approach to improving the effectiveness of existing Web search systems by means of ontologies

The main goal of this work was to enhance information retrieval by the use of ontologies. Another goal was to develop a flexible approach that can extend existing search systems with semantic technologies (i.e. make use of the advantages of existing

systems). Consequently, it was not the goal of this work to create a fully-fledged semantic search system. Therefore, we have proposed an approach that can extend existing search systems without altering the core components (i.e. indexing and ranking) of such systems. A system can be extended to utilise ontologies in a flexible manner (as described in C3). We have proven that FVs can be used to disambiguate search and hence improve the retrieval effectiveness of the search system by more than 10%. Furthermore, the prototype (Prototype I, Section 4.2.2) seemed to perform better for shorter entity-based queries than similar keyword-based queries (see Experiment I, Section 4.3.1).

C2: A flexible approach applicable to multilingual and task-driven search applications

In addition to improving the effectiveness of existing search systems (see C1), we have also proposed the application of this approach to a wider variety of different search applications. Therefore, two alternative search scenarios have been explored to test the applicability of the approach. (1) a cross-lingual information retrieval application was proposed where the query terms, with corresponding concepts, were translated directly to another language by the use of FVs. Two approaches were proposed, where the most successful approach utilised the semantic relations between the query terms in the translation process. FVs were used to achieve a 100% match, with respect to the original FV, when twice-translated. However, this was a limited scale experiment and more thorough testing needs to be done before we can conclude how successful this proposed approach is in general. (2) a scenario-driven information retrieval approach was also proposed. The approach was task driven and the IR system was extended with FVs to increase the retrieval effectiveness. We elaborated on the approach and explained how it can be integrated in rule-based systems in order to support incomplete inference, employ scalability, and the efficiency of IR engines. This approach was neither fully implemented nor tested; consequently we could not conclude on the success of this approach before it is fully tested. Nevertheless, the proposed approaches showed potential applicability of the approach described in this thesis.

C3: An unsupervised approach to associate entities from ontologies with related terminologies in textual documents

A broad literature study was conducted. Based on this study and a set of defined principles (Section 1.3) a theoretical framework was created. Two prototypes were implemented and tested.

Testing showed that our proposed approach, where every ontology entity is associated with a feature vector that is tailored to the specific terminology provided in textual documents, can improve the search results when applied to a search engine. Our prototype showed that the retrieval effectiveness could be improved by more than 10% (see Section 4.3.2). Furthermore, the approach is flexible in the sense that it is a non-supervised solution that is applicable to any ontology (i.e. there needs to be some correlation between the applied ontology and the text corpus). However, the main advantage of the approach is that a diverse corpus, like the Web, can be used since word disambiguation is handled by utilising the relationships between the ontology entities.

We have also tested the robustness of the approach with respect to changes provided by the underlying search engines (e.g. changes in text corpus or ranking of documents). Results showed that a high number of search results (i.e. more than 100) minimises the difference of using alternative search engines (paper P4), and hence changes in terminology. Furthermore, when measuring the long term Web Drift Effect (see paper P4) we found changes in terminology of less than 3% over one year (in this experiment 30 search results were used in the FVC process). A higher number of search results (i.e. 100 search results) would probably have decreased this number even further. These results combined indicate that the FVs are relatively persistent and consequently require low update frequency. However, this will probably be highly domain dependent. Nevertheless, in the case of changes to the ontologies, only FVs that are directly related to those specific parts need to be updated.

C4: A set of guidelines and parameters for optimising feature vectors with respect to ontology quality

A set of experiments was conducted to test the sensitivity of the proposed feature vector construction approach. We tested ontologies of different granularity (i.e. different number of relationships and instances) to test which features influence search performance. We found that richer ontologies (i.e. those having more relations and instances) provided better results than comparable, less rich ontologies. Furthermore, we categorised the ontologies into three distinct categories: taxonomy, lightweight and advanced. Based on these experiments and corresponding findings a set of generic guidelines on optimal parameters for FVC was proposed (listed in Section 4.3.2). We also proposed a set of guidelines specific to ontologies of different categories (also listed in Section 4.3.2).

C5: An evaluation framework for assessing feature vectors' quality with respect to both the ontology and the text corpus used

Based on a broad literature study we found that suitable evaluation methods were scarce. Furthermore, we found the approaches difficult to compare since they were often measured indirectly (i.e. their performance with respect to their designated application). We conducted an experiment where we tested the quality of the feature vectors indirectly (Experiment I, Section 4.3.1). However, in general it is difficult to find optimal settings when measuring something indirectly; consequently a more direct measure to assess the quality of FVs was needed. Therefore, a set of intrinsic (i.e. Average FV Similarity (AFVS) and Average FV Neighbourhood Similarity (AFVNS)) and extrinsic (i.e. Average FV NGD (AFVNGD)) evaluation measures were proposed. In addition, an overall FV quality score was proposed (i.e. FV Quality Score (FVQS)) which is an aggregated score of the previous three scores. We also proposed a method to measure changes provided by the search engine over time (i.e. the Web Drift Effect). When combined these measures provide a good indication of the FVs' quality with respect to the ontologies and the applied document collection. However, these measures only provide an indication of the quality, since the real value of the FVs must be viewed in the light of how they are used.

Furthermore, we have argued that FVs, or similar, are widely used in many different applications (e.g. ontology alignment, ontology mapping, semantic search, ontological

filtering). Given the wide area of use, we believe that these measures can provide useful insights into how to evaluate the quality of these FVs.

5.3 Contributions in relation to related work

In this section, we evaluate the contributions of this work with respect to the related work presented in Chapter 3. Next, each of the contributions is discussed.

C1: An approach to improving the effectiveness of existing Web search systems by means of ontologies

One of the goals of this work was to find a method to extend existing search systems with semantic technologies (i.e. ontologies). It was never the intention to develop a fully-fledged semantic search system targeting the Semantic Web (SW). Approaches utilising ontologies that extend existing search systems tend to focus on query expansion (see Section 3.2), and report on the improvement of search performance. Many other approaches focus on semantic annotation that creates a mapping between ontologies and documents (see Section 3.2), these approaches also report on improved search performance. However, a concern is whether these improvements are optimal, especially in the latter approach with respect to maintenance (see Section 1.3). Therefore, we propose an approach that is similar in stance to both approaches (i.e. query expansion and semantic annotation) but that tries to avoid some of the typical shortcomings related to these approaches. Our unsupervised approach has been tested and shown to improve the retrieval effectiveness of the search system by more than 10% (paper P6).

C2: A flexible approach applicable to multilingual and task-driven search applications

In Section 3.2 a range of approaches were presented that report on improved search performance. However, a concern is the applicability of these approaches (especially the approaches described in Section 3.2.4) since the investigation of their applicability is sparse. A goal of this work was to find an approach that is flexible, and hence applicable to a variety of search applications. Consequently, the approach was designed with flexibility in mind. To test the applicability of the approach two alternative search applications were proposed using the method proposed in this work. Nevertheless, feature vectors, or similar, are widely used in many different applications like ontology alignment, ontology mapping, semantic search, ontological filtering. Therefore, we hope that this approach can be useful in a variety of areas other than those search applications proposed as part of this work.

C3: An unsupervised approach to associate entities from ontologies with related terminologies in textual documents

Our approach to feature vector construction is an unsupervised solution that is applicable to any ontology or text corpus. However, to associate ontology entities with related terminology found in a text corpus there needs to be some correlation between them. This is not a requirement of the approach but a prerequisite to get any useful results. Approaches similar in mind have been found, but, to our knowledge, no similar approach uses ontologies (see Section 3.2.4). In any case, many approaches are

dependent on a highly relevant document collection being used while others on highly specific queries (Section 3.2.4). The main benefit of our approach is that a diverse corpus, like the Web, can be used, since the approach utilises the relationships between the ontology entities to disambiguate word senses. Furthermore, it is not dependent on highly specific constructed queries either, but experiments have shown that constructed queries that are more specific can improve the quality of the constructed FVs. Consequently, the FV quality will be highly dependent on both the quality of the ontology and the correlation of terminologies between the ontology and the text corpus (on the Web finding correlated documents are usually not a problem).

C4: A set of guidelines and parameters for optimising feature vectors with respect to ontology quality

Given the number of potential FV applications it is believed that our approach to FVC can be useful for many of these different areas. Therefore, in addition to the unsupervised approach to FV construction a set of guidelines based on lessons learned is given. These guidelines will provide useful insights into the FV construction process and, hopefully, help to find optimal FV construction parameters with respect to designated application. Furthermore, these guidelines are also believed to be useful with respect to the evaluation of FVs since no formal evaluation of FVs were found in other work, they were evaluated indirectly by results of designated applications (see Section 3.2.4 and 3.3).

C5: An evaluation framework for assessing feature vectors' quality with respect to both the ontology and the text corpus used

The number of methods for the evaluation of feature vectors, or similar, are limited (see Section 3.2.4). Typically, such approaches are evaluated indirectly by their performance in a designated application (paper P5). We also found few good ontologies that covered the standard evaluation corpora like the Text Retrieval Conference (TREC) corpus. Consequently, it is difficult, or impossible, to compare these approaches. Furthermore, a concern is whether the performance of these approaches is optimal (Strasunskas & Tomassen, 2010). Consequently, we proposed an approach to evaluate the quality of FVs more directly with respect to both the ontology and the text corpus used (Section 4.3.2). We hope that the method proposed will ease the process of finding an optimal solution and inspire others to conduct more detailed evaluations of their approaches. In any case, it is also always important to measure their performance with respect to their use.

5.4 Relevance of contributions

The contributions of this work have been divided into three research areas as seen in Figure 1.5. In this section, we discuss the relevant practical use of the contributions.

We have argued that feature vectors, or similar, are widely used in many different applications but are created differently. The contributions to *feature vector construction* aims to provide an insight into how ontology entities can be associated to terminologies found in textual documents. There are many different approaches to how these FVs can

be constructed. Therefore, two alternative algorithms have been proposed and a set of experiments has been conducted to get intrinsic insights into the construction process.

Furthermore, we have argued that most of the feature vector approaches evaluate the FV as a black box, i.e. evaluating the end-result of the system. Therefore, the contributions to *feature vector quality* evaluation aims at providing intrinsic insight into how the process of FV construction can be evaluated and the FV quality assessed. These evaluation measures are vital to find optimal solutions but also to get more insights into various quality aspects of the approach.

To show the applicability of the approach it has been applied to three potential *feature vector applications*. A focus of this work was to find a practical approach with respect to ontologies (i.e. ideally any ontology ought to be used), robustness (i.e. the frequency of updating the FVs), and flexibility (i.e. ideally the approach ought to be applicable to several different potential applications). Any ontology ought to be used; therefore, we have not restricted the approach to specific types of ontologies but provided some practical guidelines with respect to three distinct categories of ontologies. Furthermore, we found, in general, FVs to be robust with respect to changes in terminology over time, but this might be highly domain dependent (e.g. quickly changing terminology due to new or not yet established domains). The flexibility of the approach was upheld by not tailoring the approach to, for example, a specific search engine. Therefore, we hope that the approach can be applicable for many different applications in several areas. Furthermore, we hope that the intrinsic insight into the feature vector construction process and the evaluation of the FV quality can be used to evolve the approach even further to many different areas of use.

5.5 Validity discussion

In this section, we introduce the identified validity concerns. Validity concerns issues that need to be considered when evaluating the results of an experiment (i.e. the justifiability of the results). According to Wohlin et al. (2003) there are basically four categories of validity concerns: construct, internal, external, and conclusion validity. The identified validity concerns are discussed shortly below.

Construct validity

The construct validity is concerned with the relation between the theories and the observations (i.e. what is measured). For example, limited training of the test subject or poorly defined concepts can cause a threat to the construct validity.

In Experiment I (Section 4.3.1), the experimental tasks and the simulated information needs were comparable to similar real world situations. The test system was designed to be similar in use as many commonly used search systems currently found on the Web. Nevertheless, the test subject were given an introduction to the semantic search system. The case study was executed at the university. Most test subjects had extensive search experience, still most test subjects found the proposed system helpful when formulating the queries (i.e. help to formulate queries is typically more preferred among novice than advanced users).

In Experiment III (Section 4.3.3), the test subjects (colleagues at the university) were to analyse a set of provided search results, which required no interaction with an experimental search system. Therefore, the test subjects' individual computer skills were considered of minor importance and hence there was no need for training prior to conducting the experiment.

The quality of the FVs was in Experiment I measured indirectly, by how they performed in a search application. Real users assessed the perceived relevance of the search results. Consequently, the real impact of the FVs was difficult to determine. Therefore, in Experiment II the quality of the FVs was assessed directly using a set of measures (see Sections 4.2.3 and 4.3.2). The findings from Experiment II were validated with real users in Experiment III. Therefore, since the quality of the FVs was measured both indirectly and directly and in addition validated by real users the threat to the construct validity is, in this respect, considered low.

Internal validity

The internal validity is concerned with the cause and the effect relationship. For example, an unidentified confounding factor can influence on the results of an experiment. An example of an unidentified confounding factor can be a participant having a bad day but feels pressure to complete his part of the experiment. The combination of both having a bad day and pressure to complete the experiment can influence the participant's subjective evaluation. Threats to internal validity are uncontrollable and unidentified confounding factors can influence the outcome.

The fatigue effect was not considered relevant for either of the case studies. On average, the participants spent less than 2.5 hours to evaluate the results in Experiment I while about half an hour was expended on Experiment III.

In Experiment I and III, both students and colleagues were used. Therefore, a potential threat is the motivation of the test participants. Depending on their motivation they can artificially favour or disfavour the outcome of tasks. Issues that may influence the participants' motivation are student-teacher relationship, own similar competitive work, etc. The invitation of participation was submitted to general mailing lists at the university. None of the participants had a direct student-teacher relationship with the researchers. Nevertheless, some of the test subjects could have had false expectations of increased benefits by participating in the experiments that hence can have influenced their motivation.

In Experiment II (Section 4.3.2), a set of FVs was created with different FV construction parameters and the effect of the applied parameters was measured. Since a third party search engine for the Web was used, we had to minimise uncontrollable changes potentially provided by the underlying search engine. The FVs were created with default parameters at both the beginning and the end of the experiment, the differences between these two sets was set to be the standard deviation for the experiment. Therefore, it is believed that these potential changes provided by the third party search engine had a minimal impact on the results of the experiment.

For all experiments, commercial Web search engines were used. Therefore, we had no control over the ranking of the documents provided by the search engines. However, the ranking of the documents was not directly used by the proposed approach (i.e. indirectly

by selecting the top n retrieved documents). Nonetheless, since a limited number of search results are processed, few search results can have a dramatic effect on the FVs when compared to an alternative search engine or over a longer time span. Therefore, as part of the proposed FV construction guidelines (based on lessons learned) we have recommended that an adequate number of search results are used to compensate for this side effect of using the Web and commercial search engines.

External validity

The external validity is concerned with the ability to generalise the results to a scope outside this work. Ideally, research work ought to have as high an external validity as possible.

All the case studies were executed at the university. In Experiment I, the test subjects had extensive search experience (i.e. 16 out of 21 identified themselves as having extensive search experience as search users, six of them identified themselves as having a developing experience in addition to extensive search experience). Therefore, we believe that the results would be more in favour of our proposed system if more diverse test subjects were selected, since the test subjects found the proposed system helpful when formulating the queries (i.e. help to formulate queries is typically more preferred among novice than advanced users (Suomela & Kekalainen, 2005)). Still, we observed an increase in retrieval effectiveness of the proposed system compared to the baseline. While in Experiment III, the users only evaluated the provided search results and were not able to influence the submitted query. Therefore, it is believed that individual computer skills were of minor importance for both experiments.

The experiments have been conducted using only our proposed semantic search system. Nonetheless, we believe that the conclusions and lessons learned are applicable to all similar approaches, especially those using ontologies to construct FVs to be applied in a search context.

Conclusion validity

The conclusion validity is concerned with the relationship between the treatment and the outcome.

In the first experiment (Experiment I, Section 4.3.1), the quality of the feature vectors was measured indirectly by how they performed in a semantic search application. The test subjects provided subjective evaluations. The subjects needed to interpret the experimental materials and tasks according to their experience (i.e. the intention was to create an experiment close to a real world Web search experience). Evaluation of the results of the post-task questionnaire showed that the experience seemed to be similar for most of test subjects. Still, we observed a variance among the users' queries and the document relevance judgments. In this experiment, the feature vectors were evaluated indirectly and consequently difficult to assess the real impact of the FVs. This issue was addressed in Experiments II and III.

In Experiment II (Section 4.3.2) the feature vector quality was measured directly by a set of intrinsic and extrinsic quality metrics. A set of distinct FVs was created using different parameters in the FV construction process. The effect of the applied parameters was measured with respect to a set of FVs created with default parameters or

FVs created with related parameters. Since the FVs were either evaluated with respect to the baseline or related FVs, the conclusions are valid within this context. However, we also tried to map the findings and lessons learned to similar findings from Experiment I. Validation of these findings was the purpose of the next experiment (Experiment III, Section 4.3.3).

Therefore, in the third experiment (Experiment III, Section 4.3.3), the proposed quality metrics were validated with real users to validate the results from Experiment II. The test subjects were given a set of simulated information needs with corresponding retrieved documents that were evaluated subjectively. The simulated information needs were based on a set of selected entities (the entities were selected using a set of defined criteria based on assessed FV quality). The simulated information needs were designed to be basic in the sense that they reflected the submitted queries in a general manner (i.e. being just adequate). The queries were created from a set of entities selected based on a set of criteria (see Section 4.3.3). In any case, in this experiment, we observed the difference between the search results from each query and not the overall search performance of the system. Therefore, it was considered to be of minor importance if the queries were not optimal according to the preferences' for each test subject (i.e. based on the information need a test subject might have liked to submit an alternative query to the query submitted). In addition, the Cronbach's alpha scores were above 0,7 indicating reliable evaluation results.

6 Conclusions and Future Work

In this final chapter of the thesis, we conclude the work by summarising the results and provide directions for future work.

6.1 Conclusions

In this thesis, we have studied how the effectiveness of an information retrieval system can be improved by utilising ontologies. Four research questions were formulated and answered, which led to five contributions of this work published in eight scientific papers.

In the literature, we found that ontologies are used differently to improve retrieval effectiveness (e.g. query expansion, query disambiguation, navigational search). Most approaches target smaller domains by using either specific ontologies or specific text corpora, while few target the Web. Our proposed approach to semantic search targets the Web. The approach is based on a pragmatic use of ontologies by relating the ontology entities with the actual terminology used in a text corpus like the Web.

We especially focused on how domain terminology provided in ontologies can be connected with the actual terminology found in textual documents. Furthermore, part of the objective was to find an approach that is applicable to existing search systems (Section 1.4). In this work, we propose to associate (i.e. extend) every ontology entity with a feature vector to tailor them to related domain terminology in a text corpus (e.g. the Web). These FVs are used to disambiguate search.

The approach can extend an existing search system, and only depends on being able to retrieve search results from the underlying system. Two prototypes were implemented and evaluated to test the applicability of the approach. We conducted an experiment, with potential end users of such system, and found that the approach can improve the retrieval effectiveness of the underlying search system by more than 10% on average. In the first experiment, we investigated the FVs effect on search and consequently evaluated indirectly. A second experiment was conducted to directly evaluate the quality of the FVs. In this experiment, we analysed the feature vector construction algorithms and the effect on the quality of the FVs by applying alternative techniques. Based on findings and lessons learned from this experiment, we proposed a set of guidelines (with respect to ontologies of different categories) for the construction of FVs.

Frameworks for evaluation of the quality of feature vectors, or similar approaches as topic signatures, are scarce. Therefore, a set of quality measures derived from contemporary literature was proposed. The FV quality is assessed using both intrinsic and extrinsic measures with respect to the ontologies. The intrinsic measures indicate the uniqueness and neighbourhood similarity aspects of the FVs, while the extrinsic measure indicates the semantic distance between the entities and their FVs. These measures were used to assess the quality of the FVs and were validated in a third experiment. The analysis of the experiment showed that the measures provide good

indications of the quality of the FVs but need to be revised to better reflect FVs used in search. The real value of the FVs must be seen in the light of how they are used (i.e. in real applications).

We have shown how this approach can be used to disambiguate search and hence subsequently improve the retrieval effectiveness of a search system. Furthermore, we have shown the robustness of the approach. For example, it is neither dependent on highly specific constructed queries (more specific queries can improve the quality of the constructed FVs), nor on a collection of only relevant documents (a diverse corpus, like the Web, can be used), and the FVs are relatively persistent (little maintenance is required).

6.2 Directions for future work

As this research has answered the research questions raised, new research questions have also emerged from this work. Next, we will provide possible research directions for future work:

- The approach did not capture synonyms as well as initially hoped. This is mainly due to synonyms being sparsely collocated; therefore their statistical significance is low and hence difficult to capture. We have shown that the approach is capable of finding some synonyms or misspelled versions of the concepts (see paper P5). However, the algorithms were not able to identify which terms were synonyms. Ideally, the synonyms ought to be part of the ontologies and hence used when constructing the FVs. Synonyms are important to improve recall while the FVs can increase the precision by filtering the search results.
- Previous work has shown that query expansion can increase recall, but the downside is usually decreased precision (in our approach the precision is increased by FV filtering of the search results). In this approach, queries can be expanded with neighbouring ontology entities, but we have not focused on creating optimal queries like (Agirre et al., 2000). Potentially the quality of the FVs can be increased even further if a strategy for query expansion is made (e.g. use of OR, + and other typical operators).
- We used the Vector Space Model (Manning et al., 2008) as a basis to represent the FVs. It would have been interesting to test the applicability of this approach using alternative models like Fuzzy-Algebra and Probabilistic based models (Dominich, 2008). We assume the approach is suitable for any model supporting term weighting. However, this needs to be empirically confirmed.
- We found the approach performed differently for the various search tasks. Analysis of different search tasks and corresponding performance of the approach on those tasks showed that certain ontology elements had a larger effect on certain information tasks than other ontology elements. This indicates a need for further research on how the approach can be tailored to various search task categories and on seamless integration with the traditionally simple Web search interface.
- A limited number of ontologies were used in this work. The selected ontologies were of different granularity and contained, in general, concept labels considered adequate for search. A problem with many ontologies, with regards to search, is artificial naming of concepts partly because concepts need to be unique as part of an

ontology (this effect is more noticeable for larger ontologies like ISO-15926). As a result, concepts can be represented by relatively long phrases that typically do not exist as part of the terminology used in textual documents (Tomassen, 2007). Consequently, such concepts in their initial state are not suitable for search. Therefore, special adaptation of such concepts is needed both when creating associated FVs but also when used in search.

- The proposed semantic search application was implemented and tested with real users. With the use of FVs, the approach focuses on the content of documents rather than how documents relate to other documents with respect to the ranking of the search results (still only a ranked list of documents were presented to the users in the conducted experiments). Since the focus is on the content of the documents, there is a need to explore further whether the approach can be improved to retrieve content rather than references to documents (i.e. paragraphs or concatenated information).
- As part of this work, a set of metrics for FV quality assessment was proposed. However, as we concluded in paper P5, alternative approaches to aggregate the total FV quality score that better fits various search applications and search tasks need to be explored.
- We found that evaluation frameworks suitable for semantic search systems are sparse, a fact that is also acknowledged by (Wrigley et al., 2010). In (Strasunskas & Tomassen, 2010) we surveyed a set of semantic search systems and their evaluation methods. Based on the analysis and findings from contemporary literature we proposed a holistic evaluation framework for semantic search systems (QuaSIR). Wrigley et al. (2010) have also proposed an evaluation framework for semantic search tools. This issue seems to attract increasing attention and invites further exploration.

References

- Adi, T., Ewell, O.K. & Adi, P. (1999), *High Selectivity and Accuracy with READWARE's Automated System of Knowledge Organization*. Management Information Technologies, Inc. (MITi).
- Agirre, E., Ansa, O., Hovy, E.H. & Martínez, D. (2000) 'Enriching very large ontologies using the WWW'. In: Staab, S., Maedche, A., Nedellec, C. & Wiemer-Hastings, P.M. (eds), *Proc. of the First Workshop on Ontology Learning OL'2000*, 31, CEUR-WS, Berlin, Germany.
- Aitken, S. & Reid, S. (2000) 'Evaluation of an ontology-based information retrieval tool'. In: Gomez-Pérez, A., Benjamins, V.R., Guarino, N. & Uschold, M. (eds), *Proc. of Workshop on the Applications of Ontologies and Problem-Solving Methods*, Berlin, Germany.
- Alemayehu, N. (2003) 'Analysis of performance variation using query expansion'. *J. Am. Soc. Inf. Sci. Technol.*, 54 (5), pp. 379-391.
- Amaral, C., Laurent, D., Martins, A., Mendes, A. & Pinto, C. (2004) 'Design and Implementation of a Semantic Search Engine for Portuguese'. In: *Proc. of 4th Int. Conf. on Language Resources and Evaluation (LREC 2004)*, 1, Lisbon, Portugal, pp. 247-250.
- Anyanwu, K., Maduko, A. & Sheth, A. (2005) 'SemRank: Ranking Complex Relationship Search Results on the Semantic Web'. In: *Proc. of the 14th Int. Conf. on World Wide Web*, ACM, New York, USA, pp. 117-127.
- Apache (1999). *Tomcat*. [Online] Available from: <http://tomcat.apache.org/> [Accessed 13.04.2010].
- Aula, A. (2003) 'Query Formulation in Web Information Search'. In: Isafas, P. & Karmakar, N. (eds), *Proc. of the IADIS Int. Conf. WWW/Internet 2003*, 1, IADIS Press, Algarve, Portugal, pp. 403-410.
- Baeza-Yates, R. (2003) 'Information retrieval in the Web: beyond current search engines'. *Int. J. of Approximate Reasoning*, 34 (2), pp. 97-104.
- Baeza-Yates, R. & Ribeiro-Neto, B. (1999) *Modern information retrieval*, ACM Press, New York.
- Berners-Lee, T., Hendler, J. & Lassila, O. (2001) 'The Semantic Web'. *Scientific American*, 285 (5), pp. 28-37.
- Bhogal, J., Macfarlane, A. & Smith, P. (2007) 'A review of ontology based query expansion'. *Inf. Process. Manage.*, 43 (4), pp. 866-886.
- Blacoe, I., Palmisano, I., Tamma, V. & Iannone, L. (2008) 'QuestSemantics-Intelligent Search and Retrieval of Business Knowledge'. In: *Proc. of the 2008 conf. on ECAI 2008: 18th European Conference on Artificial Intelligence*, 178, IOS Press, Amsterdam, pp. 648-652.
- Bonino, D., Corno, F., Farinetti, L. & Bosca, A. (2004) 'Ontology Driven Semantic Search'. *WSEAS Transaction on Information Science and Application*, 1 (6), pp. 1597-1605.
- Borlund, P. (2009) 'User-centered Evaluation of Information Retrieval Systems'. In: Goker, A. & Davies, J. (eds), *Information Retrieval: Searching in the 21st Century*, Wiley, Chichester, UK, pp. 21-37.
- Braga, R.M.M., Werner, C.M.L. & Mattoso, M. (2000) 'Using Ontologies for Domain Information Retrieval'. In: *Proc. of the 11th International Workshop on Database and Expert Systems Applications*, IEEE Computer Society, Greenwich, London, U.K., pp. 836-840.
- Brasethvik, T. (2004) 'Conceptual modeling for domain specific document description and retrieval - An approach to semantic document modeling', NTNU, Trondheim.
- Brasethvik, T. & Gulla, J.A. (2001) 'Natural language analysis for semantic document modeling'. *Data & Knowledge Engineering*, 38 (1), pp. 45-62.
- Brickley, D. & Guha, R.V. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema - W3C Recommendation*. [Online] Available from: <http://www.w3.org/RDF/> [Accessed 27/01].
- Burton-Jones, A., Storey, V., Sugumaran, V. & Purao, S. (2003) 'A Heuristic-Based Methodology for Semantic Augmentation of User Queries on the Web'. In: *Conceptual Modeling - ER 2003*, LNCS 2813, Springer, Heidelberg, pp. 476-489.
- Carpineto, C. & Romano, G. (2010) 'Towards More Effective Techniques for Automatic Query Expansion'. In: *Research and Advanced Technology for Digital Libraries*, LNCS 1696, Springer, Heidelberg, pp. 851-852.
- Carrot2 (2009). *Carrot2 an Open Source Search Results Clustering Engine*. [Online] Available from: <http://www.carrot2.org/> [Accessed 21.02.2011].
- Castells, P., Fernandez, M. & Vallet, D. (2007) 'An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval'. *IEEE Transactions on Knowledge and Data Engineering*, 19 (2), pp. 261-272.
- Chang, Y., Kim, M. & Ounis, I. (2004) 'Construction of Query Concepts in a Document Space Based on Data Mining Techniques'. In: Christiansen, H., Hacid, M.-S., Andreasen, T. & Larsen, H.L. (eds), *Flexible Query Answering Systems*, LNCS 3055, Springer, Heidelberg, pp. 137-149.
- Chang, Y., Ounis, I. & Kim, M. (2006) 'Query reformulation using automatically generated query concepts from a document space'. *Information Processing and Management*, 42 (2), pp. 453-468.
- Chenggang, W., Wen-Pin, J., Qi-Jia, T. & Zhong-Zhi, S. (2001) 'An information retrieval server based on ontology and multi-agent'. *Journal of Computer Research and Development*, 38 (6), pp. 641-647.

- Chirita, P.-A., Costache, S., Nejd, W. & Paiu, R. (2006) 'Beagle++: Semantically Enhanced Searching and Ranking on the Desktop'. In: *The Semantic Web: Research and Applications*, LNCS 4011, Springer, Heidelberg, pp. 348-362.
- Cilibrasi, R. & Vitanyi, P. (2007) 'The Google Similarity Distance'. *IEEE Transactions on Knowledge and Data Engineering*, 19 (3), pp. 370-383.
- Ciorascu, C., Ciorascu, I. & Stoffel, K. (2003) 'knOWler - Ontological Support for Information Retrieval Systems'. In: *Proc. of SIGIR 2003 Conference, Workshop on Semantic Web*, Toronto, Canada.
- 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] Available from: <http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218> [Accessed 13/07].
- Corby, O., Dieng-Kuntz, R., Faron-Zucker, C. & Gandon, F. (2006) 'Searching the Semantic Web: Approximate Query Processing Based on Ontologies'. *IEEE Intelligent Systems*, 21 (1), pp. 20-27.
- d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. & Guidi, D. (2008) 'Toward a New Generation of Semantic Web Applications'. *IEEE Intelligent Systems*, 23 (3), pp. 20-28.
- Delone, W.H. & McLean, E.R. (2003) 'The DeLone and McLean Model of Information Systems Success: A Ten-Year Update'. *J. Manage. Inf. Syst.*, 19 (4), pp. 9-30.
- Demartini, G. & Mizzaro, S. (2006) 'A Classification of IR Effectiveness Metrics'. In: Lalmas, M., MacFarlane, A., Rüger, S., Tombros, A., Tsirikia, T. & Yavlinsky, A. (eds), *Advances in Information Retrieval*, LNCS 3936, Springer, Heidelberg, pp. 488-491.
- Desmontils, E. & Jacquin, C. (2001) 'Indexing a web site with a terminology oriented ontology'. In: *Proc. of the First Semantic Web Working Symposium (SWWS'01)*, pp. 549-565.
- Ding, L., Finin, T., Joshi, A., Peng, Y., Pan, R. & Reddivari, P. (2005) 'Search on the Semantic Web'. *IEEE Computer*, 10 (38), pp. 62-69.
- Dominich, S. (2008) *The Modern Algebra of Information Retrieval*, 24, Springer-Verlag, Heidelberg.
- Escudero, G., Marquez, L. & Rigau, G. (2000) 'A comparison between supervised learning algorithms for word sense disambiguation'. In: *Proc. of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, Association for Computational Linguistics, Lisbon, Portugal, pp. 31-36.
- Esmaili, K.S. & Abolhassani, H. (2006) 'A Categorization Scheme for Semantic Web Search Engines'. In: *Proc. of the IEEE Int. Conf. on Computer Systems and Applications*, pp. 171-178.
- Fellbaum (1998) *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*, The MIT Press.
- Formica, A., Missikoff, M., Pourabbas, E. & Taglino, F. (2008) 'Weighted Ontology for Semantic Search'. In: Meersman, R. & Tari, Z. (eds), *On the Move to Meaningful Internet Systems: OTM 2008*, LNCS 5332, Springer, Heidelberg, pp. 1289-1303.
- Frank, A., Krieger, H.-U., Xu, F., Uszkoreit, H., Crysmann, B., Jorg, B. & Schafer, U. (2007) 'Question answering from structured knowledge sources'. *J. of Applied Logic*, 5 (1), pp. 20-48.
- Gabrilovich, E. & Markovitch, S. (2005) 'Feature Generation for Text Categorization Using World Knowledge'. In: *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, USA, pp. 1048-1053.
- Gabrilovich, E. & Markovitch, S. (2007) 'Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization'. *J. Mach. Learn. Res.*, 8 (pp. 2297-2345).
- Gabrilovich, E. & Markovitch, S. (2009) 'Wikipedia-based Semantic Interpretation for Natural Language Processing'. *J. of Artificial Intelligence Research*, 34 (pp. 443-498).
- Gao, X., Murugesan, S. & Lo, B. (2004) 'Multi-Dimensional Evaluation of Information Retrieval Results'. In: *Proc. of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, IEEE Computer Society, Washington, DC, USA, pp. 192-198.
- Google (2009). *Google AJAX Search API*. [Online] Available from: <http://code.google.com/apis/ajaxsearch/> [Accessed 21.02.2011].
- Griffiths, J., Johnson, F. & Hartley, R. (2007) 'User satisfaction as a measure of system performance'. *J. of Librarianship and Information Science*, 39 (3), pp. 142-152.
- Grootjen, F.A. & van der Weide, T.P. (2006) 'Conceptual query expansion'. *Data & Knowledge Engineering*, 56 (2), pp. 174-193.
- Gruber, T.R. (1993) 'A translation approach to portable ontology specifications'. *Knowledge Acquisition*, 5 (2), pp. 199-220.
- Guha, R., McCool, R. & Miller, E. (2003) 'Semantic Search'. In: *Proc. of the 12th int. conf. on World Wide Web*, ACM, New York, USA, pp. 700-709.
- Gulla, J.A., Tomassen, S.L. & Strasunskas, D. (2006) 'Semantic Interoperability in the Norwegian Petroleum Industry'. In: Karagiannis, D. & Mayer, H.C. (eds), *Proc. of the 5th International Conference on Information Systems Technology and its Applications (ISTA 2006)*, P-84, Lecture Notes in Informatics (LNI) Köllen Druck+Verlag GmbH, Bonn, Klagenfurt, Austria, pp. 81-94.
- Gyongyi, Z. & Garcia-Molina, H. (2005) 'Web Spam Taxonomy'. In: *First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb 2005)*.

- Harter, S.P. (1992) 'Psychological relevance and information science'. *Journal of the American Society for Information Science*, 43 (9), pp. 602-615.
- Harter, S.P. (1996) 'Variations in relevance assessments and the measurement of retrieval effectiveness'. *Journal of the American Society for Information Science*, 47 (1), pp. 37-49.
- Harth, A., Hogan, A., Delbru, R., Umbrich, J., O'Riain, S. & Decker, S. (2007) 'SWSE: Answers Before Links!'. In: Golbeck, J. & Mika, P. (eds), *Proc. of the Semantic Web Challenge 2007*, 295, CEUR-WS, Busan, Korea.
- Horrocks, I. (2007) 'Semantic web: the story so far'. In: *Proc. of the 2007 int. cross-disciplinary conf. on Web accessibility (W4A)*, ACM, Banff, Canada, pp. 120-125.
- Huffman, S.B. & Hochster, M. (2007) 'How well does result relevance predict session satisfaction?'. In: *Proc. of the 30th annual int. ACM SIGIR conf. on Research and development in information retrieval*, ACM, Amsterdam, The Netherlands, pp. 567-574.
- ISO (2007). *ISO/TS 15926-4:2007 Industrial automation systems and integration -- Integration of life-cycle data for process plants including oil and gas production facilities -- Part 4: Initial reference data*. [Online] Available from: http://www.iso.org/iso/catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41329 [Accessed 15022011].
- Jiang, X. & Tan, A.-H. (2006) 'OntoSearch: A Full-Text Search Engine for the Semantic Web'. In: *Proc. of the 21st Nat. Conf. on Artificial Intelligence*, 2, AAAI Press, Boston, Massachusetts, pp. 1325-1330.
- Kim, H.H. (2005) 'ONTOWEB: Implementing an ontology-based Web retrieval system'. *J. of the American Society for Information Science and Technology*, 56 (11), pp. 1167-1176.
- Kiryakov, A., Popov, B., Terziev, I., Manov, D. & Ognyanoff, D. (2004) 'Semantic annotation, indexing, and retrieval'. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2 (1), pp. 49-79.
- Kobayashi, M. & Takeda, K. (2000) 'Information retrieval on the web'. *ACM Computing Surveys*, 32 (2), pp. 144-173.
- Kulkarni, S. & Caragea, D. (2009) 'Towards Bridging the Web and the Semantic Web'. In: *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2009. WI-IAT '09*, IEEE Computer Society, Milano, Italy, pp. 667-674.
- Laclavik, M., Seleng, M., Gatial, E., Balogh, Z. & Hluchy, L. (2007) 'Ontology based Text Annotation --OnTeA'. In: *Proc. of the 2007 conf. on Information Modelling and Knowledge Bases XVIII*, IOS Press, pp. 311-315.
- Lewandowski, D. (2005) 'Web searching, search engines and Information Retrieval'. *Information Services and Use*, 25 (3), pp. 137-147.
- Lopez, V., Motta, E. & Uren, V. (2006a) 'PowerAqua: Fishing the Semantic Web'. In: *The Semantic Web: Research and Applications*, LNCS 4011, Springer, Heidelberg, pp. 393-410.
- Lopez, V., Sabou, M. & Motta, E. (2006b) 'PowerMap: Mapping the Real Semantic Web on the Fly'. In: *The Semantic Web - ISWC 2006*, LNCS 4273, Springer, Heidelberg, pp. 414-427.
- Lopez, V., Uren, V., Motta, E. & Pasin, M. (2007) 'AquaLog: An ontology-driven question answering system for organizational semantic intranets'. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5 (2), pp. 72-105.
- Mangold, C. (2007) 'A survey and classification of semantic search approaches'. *Int. J. Metadata, Semantics and Ontologies*, 2 (1), pp. 23-34.
- Manning, C.D., Raghavan, P. & Schütze, H. (2008) *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK.
- McCool, R., Cowell, A. & Thurman, D. (2005) 'End-User Evaluations of Semantic Web Technologies'. In: Bernstein, A., Androutsopoulos, I., Degler, D. & McBride, B. (eds), *Proc. of the ISWC 2005 Workshop on End User Semantic Web Interaction*, 172, CEUR-WS, Galway, Ireland, p. 7.
- McGuinness, D.L. & van Harmelen, F. (2004). *OWL Web Ontology Language - W3C Recommendation*. [Online] Available from: <http://www.w3.org/2004/OWL/> [Accessed 21.01.2011].
- Melchuk, I. (1981) 'Meaning-text models: a recent trend in Soviet linguistics'. *Annual Review of Anthropology*, 10 (pp. 27-62).
- Nagypal, G. (2005) 'Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies'. In: Meersman, R., Tari, Z. & Herrero, P. (eds), *On the Move to Meaningful Internet Systems 2005: OTM Workshops*, LNCS 3762, Springer, Heidelberg, pp. 780-789.
- Nagypal, G. (2007) 'Possibly imperfect ontologies for effective information retrieval', PhD thesis, University of Karlsruhe, Germany.
- Navigli, R. (2009) 'Word Sense Disambiguation: A Survey'. *ACM Comput. Surv.*, 41 (2), pp. 1-69.
- Niles, I. & Pease, A. (2001) 'Towards a standard upper ontology'. In: *Proc. of the international conference on Formal Ontology in Information Systems - Volume 2001*, ACM, New York, USA, pp. 2-9.
- Ning, X., Jin, H., Jia, W. & Yuan, P. (2009) 'Practical and effective IR-style keyword search over semantic web'. *Information Processing & Management*, 45 (2), pp. 263-271.
- Ogden, C.K. & Richards, I.A. (1927) *The Meaning of Meaning: A Study of The Influence of Language upon Thought and of the Science of Symbolism*, Kegan Paul, Trench, Trubner & Co, London.
- Ozcan, R. & Aslangöğün, Y.A. (2004), *Concept Based Information Access Using Ontologies and Latent Semantic Analysis*. University of Texas at Arlington.

- Page, L., Brin, S., Motwani, R. & Winograd, T. (1999), *The PageRank Citation Ranking: Bringing Order to the Web*. Stanford InfoLab.
- Pan, J., Thomas, E. & Sleeman, D. (2006) 'ONTOSEARCH2: Searching and querying web ontologies'. In: *Proc. of the IADIS Int. Conf.*, pp. 211-218.
- Panagis, Y., Sakkopoulos, E., Garofalakis, J. & Tsakalidis, A. (2006) 'Optimisation mechanism for web search results using topic knowledge'. *Int. J. of Knowledge and Learning*, 2 (pp. 140-153).
- Paralic, J. & Kostial, I. (2003) 'Ontology-based Information Retrieval', *Information and Intelligent Systems*, Croatia, pp. 23-28.
- Passmore, C., Dobbie, A.E., Parchman, M. & Tysinger, J. (2002) 'Guidelines for Constructing a Survey'. *Family Medicine Journal*, 24 (4), pp. 281-286.
- Phillips, E.M. & Pugh, D.S. (2005) *How to Get a PhD: A Handbook For Students And Their Supervisors*, Open University Press, Buckinghamshire.
- Piwowski, B., Gallinari, P. & Dupret, G. (2007) 'Precision recall with user modeling (PRUM): Application to structured information retrieval'. *ACM Trans. Inf. Syst.*, 25 (1), p. 1.
- Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D. & Goranov, M. (2003) 'KIM - Semantic Annotation Platform'. In: Fensel, D., Sycara, K.P. & Mylopoulos, J. (eds), *The SemanticWeb - ISWC 2003*, LNCS 2870, Springer, Heidelberg, pp. 834-849.
- Qiu, Y. & Frei, H.-P. (1993) 'Concept based query expansion'. In: *Proc. of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, Pittsburgh, Pennsylvania, United States, pp. 160-169.
- Rajapakse, R. & Denham, M. (2002) 'Concept Based Adaptive IR Model Using FCA-BAM Combination for Concept Representation and Encoding'. In: Crestani, F., Girolami, M. & van Rijsbergen, C.J. (eds), *Advances in Information Retrieval*, LNCS 2291, Springer, Heidelberg, pp. 79-84.
- Rajapakse, R.K. & Denham, M. (2006) 'Text retrieval with more realistic concept matching and reinforcement learning'. *Information Processing & Management*, 42 (5), pp. 1260-1275.
- Reeve, L. & Han, H. (2005) 'Survey of semantic annotation platforms'. In: *Proc. of the 2005 ACM symposium on Applied computing*, ACM, Santa Fe, New Mexico, pp. 1634-1638.
- Rehbein, I., Ruppenhofer, J. & Sunde, J. (2009) 'MaJo - A Toolkit for Supervised Word Sense Disambiguation and Active Learning'. In: Passarotti, M., Przepiórkowski, A., Raynaud, S. & Eynde, F.V. (eds), *Proc. of the Eighth International Workshop on Treebanks and Linguistic Theories*, EDUCatt, Milan, Italy, pp. 161-172.
- Rocha, C., Schwabe, D. & Aragao, M. (2004) 'A Hybrid Approach for Searching in the Semantic Web'. In: *Proc. of the 13th international conference on World Wide Web*, ACM, New York, USA, pp. 374-383.
- Sandsmark, N. & Mehta, S. (2004) 'Integrated Information Platform for Reservoir and Subsea Production Systems'. In: *Proc. of the 13th Product Data Technology Europe Symposium (PDT 2004)*, Stockholm, p. 9.
- Scheir, P., Pammer, V. & Lindstaedt, S.N. (2007) 'Information Retrieval on the Semantic Web - Does it exist?', paper presented to the LWA.
- Schumacher, K., Sintek, M. & Sauermann, L. (2008) 'Combining Fact and Document Retrieval with Spreading Activation for Semantic Desktop Search'. In: *The Semantic Web: Research and Applications*, LNCS 5021, Springer, Heidelberg, pp. 569-583.
- Sebastiani, F., Sperduti, A. & Valdambrini, N. (2000) 'An Improved Boosting Algorithm and its Application to Text Categorization'. In: *Proc. of 9th Int. Conf. on Information and Knowledge Management*, ACM, New York, USA, pp. 78-85.
- Simperl, E., Popov, I. & Bürger, T. (2009) 'ONTOCOM Revisited: Towards Accurate Cost Predictions for Ontology Development Projects'. In: pp. 248-262.
- Sjøberg, D., Anda, B., Arisholm, E., Dybå, T., Jørgensen, M., Karahasanović, A. & Vokác, M. (2003) 'Challenges and Recommendations When Increasing the Realism of Controlled Software Engineering Experiments'. In: pp. 24-38.
- Solskinnsbakk, G. & Gulla, J. (2008a) 'Ontological Profiles as Semantic Domain Representations'. In: Kapetanios, E., Sugumaran, V. & Spiliopoulou, M. (eds), *Natural Language and Information Systems*, LNCS 5039, Springer, Heidelberg, pp. 67-78.
- Solskinnsbakk, G. & Gulla, J. (2008b) 'Ontological Profiles in Enterprise Search'. In: Gangemi, A. & Euzenat, J. (eds), *Knowledge Engineering: Practice and Patterns*, LNCS 5268, Springer, Heidelberg, pp. 302-317.
- Song, J.F., Zhang, W.M., Xiao, W.D., Li, G.H. & Xu, Z.N. (2005) 'Ontology-Based Information Retrieval Model for the Semantic Web'. In: *Proc. of the 2005 IEEE Int. Conf. on e-Technology, e-Commerce and e-Service (EEE'05)*, IEEE Computer Society, Washington, DC, USA, pp. 152-155.
- Spink, A. (2002) 'A user-centered approach to evaluating human interaction with Web search engines: an exploratory study'. *Information Processing & Management*, 38 (3), pp. 401-426.
- Standards Norway (2004) *Well integrity in drilling and well operations*, Standards Norway, p. 158<http://www.npd.no/Global/Norsk/5%20-%20Regelverk/Skjema/Br%C3%B8nregistrering/Norsk_standard_D-010.pdf>.
- Stojanovic, N., Studer, R. & Stojanovic, L. (2003) 'An Approach for the Ranking of Query Results in the Semantic Web'. In: *The Semantic Web - ISWC 2003*, LNCS 2870, Springer, Heidelberg, pp. 500-516.

- Strasunskas, D. & Tomasgard, A. 2010, 'A Method to Assess Value of Integrated Operations', *AMCIS 2010 Proceedings*. [Online] Available from: <http://aisel.aisnet.org/amcis2010/459/> [Accessed 21.02.2011].
- Strasunskas, D. & Tomassen, S.L. (2008a) 'Empirical Insights on a Value of Ontology Quality in Ontology-driven Web Search'. In: Meersman, R. & Tari, Z. (eds), *On the Move to Meaningful Internet Systems: OTM 2008*, LNCS 5332, Springer, Heidelberg, pp. 1319-1337.
- Strasunskas, D. & Tomassen, S.L. (2008b) 'The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation'. *Int. J. Knowledge and Learning*, 4 (4), pp. 398-414.
- Strasunskas, D. & Tomassen, S.L. (2010) 'On Variety of Semantic Search Systems and Their Evaluation Methods'. In: *Proc. of the Int. Conf. on Information Management and Evaluation (ICIME)*, Academic Conferences Publishing, Cape Town, South Africa, pp. 380-387.
- Su, L.T. (1992) 'Evaluation measures for interactive information retrieval'. *Information Processing & Management*, 28 (4), pp. 503-516.
- Su, X. & Gulla, J.A. (2006) 'An information retrieval approach to ontology mapping'. *Data & Knowledge Engineering*, 58 (1), pp. 47-69.
- Sullivan, D. (2002). *Death Of A Meta Tag*. [Online] Available from: <http://searchenginewatch.com/2165061> [Accessed 27/01].
- Suomela, S. & Kekalainen, J. (2005) 'Ontology as a Search-Tool: A Study of Real Users' Query Formulation With and Without Conceptual Support'. In: *Advances in Information Retrieval*, LNCS 3408, Springer, Heidelberg, pp. 315-329.
- Swoogle (2005). *Swoogle*. [Online] Available from: <http://swoogle.umbc.edu/> [Accessed 21.02.2011].
- Tablan, V., Damjanovic, D. & Bontcheva, K. (2008) 'A Natural Language Query Interface to Structured Information'. In: *The Semantic Web: Research and Applications*, LNCS 5021, Springer, Heidelberg, pp. 361-375.
- Tomassen, S.L. (2007) 'Pros & Cons of Applying Industrial Ontologies in Information Retrieval'. In: Strasunskas, D., Rao, J. & Hakkarainen, S. (eds), *Proc. of the 1st International Workshop on Semantic Technology Adoption in Business (STAB'07)*, Tapir Akademisk Forlag, Vienna, Austria, pp. 39-44.
- Ungrangsi, R., Anutariya, C. & Wuwongse, V. (2007) 'SQORE-Based Ontology Retrieval System'. In: *Database and Expert Systems Applications*, LNCS 4653, Springer, Heidelberg, pp. 720-729.
- Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. & Ciravegna, F. (2006) 'Semantic annotation for knowledge management: Requirements and a survey of the state of the art'. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4 (1), pp. 14-28.
- van Harmelen, F. (2006) 'Semantic Web Research Anno 2006: Main Streams, Popular Fallacies, Current Status and Future Challenges'. In: Klusch, M., Rovatsos, M. & Payne, T. (eds), *Cooperative Information Agents X*, 4149, LNCS Springer, Heidelberg, pp. 1-7.
- Vaughan, L. (2004) 'New measurements for search engine evaluation proposed and tested'. *Information Processing & Management*, 40 (4), pp. 677-691.
- Venkatesh, V., Morris, M., Davis, G. & Davis, F. (2003) 'User Acceptance of Information Technology: Toward a Unified View'. *Management Information Systems Quarterly*, 27 (3), p. 5.
- Voorhees, E. & Harman, D. (2005) *TREC: Experiment and evaluation in information retrieval*, MIT Press, Cambridge, MA.
- W3C (2001). *Semantic Web*. [Online] Available from: <http://www.w3.org/2001/sw/> [Accessed 21.02.2011].
- Wang, H., Zhang, K., Liu, Q., Tran, T. & Yu, Y. (2008) 'Q2Semantic: A Lightweight Keyword Interface to Semantic Search'. In: *The Semantic Web: Research and Applications*, LNCS 5021, Springer, Heidelberg, pp. 584-598.
- Wang, Y. & Forgionne, G. (2008) 'Testing a decision-theoretic approach to the evaluation of information retrieval systems'. *Journal of Information Science*, 34 (6), pp. 861-876.
- Wohlin, C., Höst, M. & Henningson, K. (2003) 'Empirical Research Methods in Software Engineering'. In: *Empirical Methods and Studies in Software Engineering*, LNCS 2765, Springer, Heidelberg, pp. 7-23.
- Wrigley, S.N., Reinhard, D., Elbedweihy, K., Bernstein, A. & Ciravegna, F. (2010) 'Methodology and Campaign Design for the Evaluation of Semantic Search Tools'. In: *Proc. of the Semantic Search 2010 Workshop (SemSearch 2010)*, ACM, New York, USA.
- Yahoo (2009). *Yahoo! Developer Network*. [Online] Available from: <http://developer.yahoo.com>.
- Yang, H.-C. (2006) 'A method for automatic construction of learning contents in semantic web by a text mining approach'. *Int. J. of Knowledge and Learning*, 2 (pp. 89-105).
- Zhang, L., Yu, Y., Zhou, J., Lin, C. & Yang, Y. (2005) 'An enhanced model for searching in semantic portals'. In: *Proc. of the 14th international conference on World Wide Web*, ACM, Chiba, Japan, pp. 453-462.
- Zhou, X., Hu, X. & Zhang, X. (2007) 'Topic Signature Language Models for Ad hoc Retrieval'. *IEEE Transactions on Knowledge and Data Engineering*, 19 (9), pp. 1276-1287.

Part II

Papers

P1: Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect

Publication details

Tomassen, S.L. & Strasunskas, D. (2009) Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect. In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, IEEE Computer Society, Washington, pp. 133-138.

Construction of Ontology based Semantic-Linguistic Feature Vectors for Searching: the Process and Effect

Stein L. Tomassen

Dept. of Computer and Information Science,
Norwegian University of Science and
Technology (NTNU), Norway
+ 47 735 94218

stein.l.tomassen@idi.ntnu.no

Darijus Strasunskas

Dept. of Industrial Economics and
Technology Management,
Norwegian University of Science and
Technology (NTNU), Norway
+47 735 93659

darijuss@gmail.com

ABSTRACT

Search is among the most frequent activities on the Web. However, the search activity still requires extra efforts in order to get satisfactory results. One of the reasons is heterogeneous information resources and exponential increase of information. The problem of heterogeneity arises as result of discipline specific language used even in the domain specific documents. This particular problem we tackle in this paper. We propose an approach to construct semantic-linguistic feature vectors that are used in search. The feature vectors are built based on domain semantics encoded in an ontology and enhanced by a relevant terminology from the actual documents on the Web. Semantic information from the ontologies is also used to expand the user queries and the feature vectors are used to filter and rank the retrieved documents. The value of this approach is twofold. First, it captures relevant semantics from an ontology, and second, it accounts for statistically significant collocations of other terms and phrases in relation to the ontology entities. In this paper, we explain how these feature vectors are constructed and what effect they have on search performance.

Categories and Subject Descriptors

H.3.3 [Information Storage And Retrieval]:
Information Search and Retrieval - *information filtering, selection process.*

General Terms

Algorithms, Experimentation.

Keywords

Information Retrieval, Ontology, Web Search, Feature Vector Construction, Evaluation.

1. INTRODUCTION

Nowadays, the Web is one of the dominant information sources for learning and acquiring new knowledge. However, finding the relevant information is still a huge challenge. The emerging Semantic Web will eventually solve some of the problems. However, we need to improve search now. Improvement can be

achieved by combining strengths of the current Web search with the emerging semantic techniques. This endeavor is referred to as semantic search that is differentiated from the querying of the Semantic Web. There are many different approaches emerging in this research area. Some approaches are relying on semantic annotations (e.g., [2, 19]) by adding additional metadata (e.g. [9]); some are enhancing clustering of retrieved documents according to topic (e.g. [14]); some are developing powerful querying languages (e.g. [5]). Therefore, there are a lot of efforts devoted to research on improvement of information retrieval (IR) by the help of ontologies that encode domain knowledge (e.g. [6, 18]).

The objective of this paper to elaborate on the proposed approach to semantic search that builds on a concept of *feature vector* (fv). The approach is based on pragmatic use of ontologies by relating the concepts (domain semantics) with the actual terminology used in text corpora. The ontologies represent the domains of interest. The general idea is that these ontologies can be used in search to avoid ambiguity of concepts. Therefore, every entity (classes and individuals) of the ontology are associated with a feature vector to tailor them to the specific terminology of the text corpora (the Web). These fvs are next used to filter and re-rank the search results from the underlying search system. We explain how these feature vectors are constructed and provide an example of the construction process. Then we present an experiment conducted and discuss how the usage of these feature vectors influence on the search quality.

This paper is organized as follows. In section 2, related work is discussed. In section 3, the algorithm of how the feature vectors are constructed is presented. In Section 4 we present an experiment and discuss some of the results. Finally, in section 5, we conclude this paper.

2. RELATED WORK

In this section we explore some related work on enhancement of search. This overview is limited to the approaches that endeavor improvement by employing analysis of semantics rather than by taking different measures. By different measures we mean analysis of

Web content w.r.t. information quality often used for ranking purposes in order to improve precision (the approaches as PageRank based on references among Web pages, ranking based on information update, etc.).

Many approaches enhance traditional vector space model by adding processing of semantics. Some start with semantic querying using ontology query languages and use resulting instances to retrieve relevant documents using vector space model [6]. Whereas Nagypal [11] combines ontology usage with vector-space model by extending a non-ontological query. There, ontology is used to disambiguate queries. Simple text search is run on the concepts' labels and users are asked to choose the proper term interpretation. Ozcan et al. [13] are using ontologies for the representation of concepts. The concepts are extended with similar words using a combination of Latent Semantic Analysis (LSA) and WordNet¹. Testing done shows promising results for short or poorly formulated queries. Braga et al. [3] are using ontologies for retrieval and filtering of domain information across multiple domains. However, the ontology usage is limited to hypernyms (super class), hyponyms (sub class), and synonyms.

Finally, the approach by Solskinnsbakk and Gulla [15] is relying on constructing ontological profiles that contain concept vectors. However, when creating the concept vectors they are depended on a highly relevant document collection. Furthermore, they also need a collection of non-relevant document in order to construct negative concept vectors. Both vectors are used for query expansion. Testing shows good results for situations where recall is more critical than precision [16].

3. FEATURE VECTOR

Every ontology entity has an associated feature vector with a set of relevant terms extracted from the document collection. An ontology entity can either be a class or an individual. In this approach we use the term entity instead of concept because a concept is often a synonym for a class when it comes to ontologies. Our approach associates feature vectors to both classes and individuals and are therefore, for the sake of easiness, referred to as entities. In this section, we will describe the process of how these f_v s are constructed, but first a definition of a feature vector is provided. At the end of this section an example of the construction process is presented.

3.1 Definition of a Feature Vector

The development of the approach is inspired by a linguistics method for describing the meaning of objects - the semiotic triangle [6]. In our approach, a feature vector "connects" a concept to a document collection, i.e., the feature vector is tailored to the specific terminology used in a particular collection of the documents. Feature vectors are built considering both semantics encoded in an ontology and a vital and dominant lexical terminology surrounding the entities

in a text corpora. Therefore, a feature vector constitutes a rich representation of the entities and is related to actual terminology used on the Web. Correspondingly, a feature vector of an entity e is represented as a two-tuple as described in the following definition:

Definition 1: Feature Vector

$$FV_e = \langle S_e, L_e \rangle \quad (1)$$

where S_e is a semantic enrichment part of FV_e that represents a set of neighborhood entities and properties in an ontology O . L_e is a linguistic enrichment of an entity that is a set of terms with a significant proximity to an entity and its semantic neighborhood. The process of selecting relevant entities and terms into these sets is elaborated in the following subsection.

3.2 Feature Vector Construction

The Feature Vector Construction algorithm is depicted in Figure 1. The algorithm constitutes three phases (main steps). The first phase includes ranking of the ontology entities according to their importance w.r.t. the ontology, this helps to optimize the feature vector construction done in phase 3. The main aim of the next phase is to extract and group sets of candidate terms being relevant to each entity. However, the candidate terms are not necessarily relevant to the domain defined by the ontology. Consequently, the aim of the last phase is to identify those candidate terms being relevant to the entities defined by the ontology. Finally, an f_v for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding f_v s that consist of terms associated both to the entities and the domain terminology.

```

Input: An ontology
Output: A feature vector for each entity of the input ontology

ONT = the ontology
EN = {e1, ..., en}, the entities of the ontology
RES = {d1, ..., dj}, a set of retrieved documents
KW = {k1, ..., kj}, a set of extracted keywords
CLU = {c1, ..., cj}, a set of clusters

Initialize rankedEntityList;
Initialize entityResultContainer;
FOR each ei ∈ EN
  score = CALL calculateEntityCentralityScore(ONT, ei); // Step 1
  rankedEntityList.addCentralityScore(ei, score);
ENDFOR
Sort rankedEntityList; // sorted by score
FOR each ei ∈ rankedEntityList
  query = CALL createEntityQuery(ONT, ei); // Step 2.1
  RES = CALL search(query); // Step 2.2
  entityResultContainer.addSearchResults(ei, RES);
  FOR each dj ∈ RES
    KW = CALL extractKeywords(dj, query); // Step 2.3
    entityResultContainer.addPageKeywords(ei, dj, KW);
  ENDFOR
  CLU = CALL cluster(ei, entityResultContainer); // Step 2.4
  entityResultContainer.addEntityClusters(ei, CLU);
ENDFOR
FOR each ei ∈ rankedEntityList
  CLU = entityResultContainer.getEntityClusters(ei);
  highestRelevance = 0;
  cl = null;
  FOR each cj ∈ CLU
    relevance = CALL calculateClusterRelevance(ONT, ei, cj, entityResultContainer); // Step 3.1
    IF relevance > highestRelevance THEN
      highestRelevance = relevance;
      cl = cj;
    ENDIF
  ENDIF
  IF cl <- null THEN
    entityFeatureVector = CALL createEntityFeatureVector(ONT, ei, cl, entityResultContainer); // Step 3.2
    entityResultContainer.addEntityFeatureVector(ei, entityFeatureVector);
  ENDIF
ENDFOR

```

Figure 1. The Feature Vector Construction algorithm.

Before the process can start an ontology needs to be selected. Below, we elaborate each of the steps as follows.

¹ WordNet, <http://wordnet.princeton.edu/>

Step 1: Rank entities

Since we endeavor to create fvs for every entity (recall that an entity can both be a class and an individual) in the ontology, the algorithm starts with traversing the ontology and ranks each entity according to relevancy. The result of this process is a ranked list of entities according to considered importance (centrality) w.r.t. the ontology. This list of ranked entities is later used to identify those documents being relevant to the domain defined by the ontology (Step 3). By using a ranked list versus a random list of entities helps to improve the quality of identifying the most relevant candidate terms done in Step 3. The idea is that more information is available for the most central entities and consequently will be better candidates to discriminate relevant candidate terms. Those entities that already have been assigned relevant terms are later used to identify the most relevant candidate terms for other entities (more of this in Step 3).

We have adapted the AKTiveRank algorithm by Alani et al. [1] to rank the entities. The original intention of AKTiveRank is to rank several ontologies for comparison. However, some of the measures are suitable to measure the centrality of entities w.r.t. the ontology. Consequently, we have focused on those elements of the algorithm, which are the class betweenness measure being part of the BETWEENNESS Measure (BEM) and the class density measure being part of the Density Measure (DEM). BEM gives an indication of the centrality of an entity in the sense of where it is graphically located within an ontology. The centrality is found by calculating the number of shortest paths that pass through each entity of the ontology. Our definition of Entity Betweenness Measure (EBM) is equal to the $bem(c)$ of BEM by Alani et al. [1] but for the sake of easiness it is presented below with our terminology as follows:

Definition 2: Entity Betweenness Measure (EBM)

Let $e_i, e_j \in \{E[O]\}$, e_i and e_j are any two entities in the ontology O . $E[o]$ is the set of entities in ontology o .

$$EBM(e) = \sum_{e_i \neq e_j \neq e \in E[o]} \frac{\sigma_{e_i e_j}(e)}{\sigma_{e_i e_j}} \quad (2)$$

where $EBM(e)$ is the Entity Betweenness Measure for entity e . $\sigma_{e_i e_j}$ is the shortest path from e_i to e_j , and $\sigma_{e_i e_j}(e)$ is the number of shortest paths from e_i to e_j that passes through e .

For the Entity Density Measure (EDM) we have adopted the class density measure by Alani et al. [1]. For simplicity reasons the our definition is provided below:

Definition 3: Entity Density Measure (EDM)

Let $S = \{S_1, S_2, S_3, S_4, S_5\} = \{subclasses[e], superclasses[e], individuals[e], siblings[e], relations[e]\}$

$$EDM(e) = \sum_{j=1}^5 w_j |S_j| \quad (3)$$

where $EDM(e)$ is the Entity Density Measure for entity e and w_j is a weight factor with default value of 1.

Then the total Entity Centrality Score (ECS) for an entity is calculated once both EBM and EDM are calculated for that class, which is the sum of those measures as seen in Definition 4.

Definition 4: Entity Centrality Score (ECS)

$$ECS(e) = \alpha EBM(e) + \beta EDM(e) \quad (4)$$

where $ECS(e)$ is the Entity Centrality Score for entity e and α and β are the centrality and density weight factors respectively. Both α and β is set to a default value of 0.5.

A pre-ranking of the entities is achieved when ECS is calculated for all the entities of the ontology. However, the current order of this list does not guarantee that e.g. the second entity is directly connected to the first entity. Hence, we would like a ranked list of entities that is based on both centrality but also where each next entity in the list directly connects with any of the prior entities of the list. This sought list of ranked entities is assured by using the Spreading Activation² algorithm. First, the entity with the highest ECS is selected. If there are several entities with equal score then the sum of the neighbor entities' ECS is calculated. The entity with the highest score is selected. The selected entity will act as the initial node of the Spreading Activation algorithm. The Spreading Activation algorithm ends when there are no more entities left in the pre-ranked list. Entities with no direct relation(s) to other entities will be omitted since those entities are identified as loners (e.g. an entity only being a subclass of `owl:Thing`). This feature vector construction algorithm is not able to associate feature vectors for loners since neighboring entities are vital in the process of identifying highly relevant terms (more of this in Step 3). The result of this step is a ranked list of entities that is based on both centrality and density of the ontology.

Step 2: Search and cluster

This step constitutes four sub-steps where the aim is to extract and group sets of candidate terms being relevant to each entity for further processing done in the final step, Step 3.

Step 2.1: Create entity query

In this step, a search query is prepared for each entity while the actual search is performed in the Step 2.2. The query is based on the entity name and expanded with relevant neighboring entities dependent on the search task used. The different search tasks are fact-finding, exploratory, and comprehensive (more information about search tasks are found in [17]). The motivation behind expanding the initial query with neighboring entities is to create a query that reflects both the ontology itself but also how each entity is

² Spreading activation,

http://en.wikipedia.org/wiki/Spreading_activation

related to other relevant entities by their closest neighbors.

Step 2.2: Entity based search

The query for each entity created in Step 2.1 is used to retrieve candidate documents for each entity. Any search engine can be used in this step. However, currently Yahoo!³ and Google⁴ (for searching in Web documents) and Nutch⁵ (for searching in local documents) are implemented. In the first experiment described in section 4 Yahoo was used. The retrieval session is keyword-based.

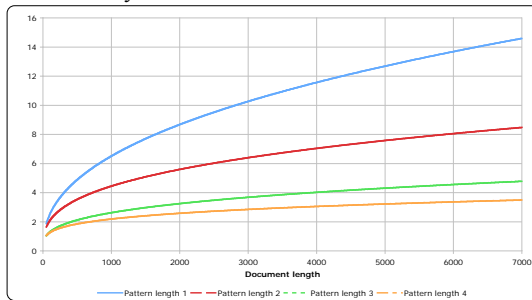


Figure 2. Term frequency threshold vs. document length.

Step 2.3: Contextual keyword extraction

For each document a set of keywords is extracted. First, a part of speech (POS) tagger is used to tag the content (snippet or full text). Currently Stanford POS Tagger⁶ and FastTag⁷ are implemented, though the latter is preferred as being faster and more effective. Then a set of tagging rules (42 rules found in Appendix A1), inspired by Justeson and Katz [8], is applied. Based on these rules a set of candidate keyphrases and keywords are extracted, hereinafter referred to as keyphrases. However, only those keyphrases that are within what we call a contextual window are extracted. A contextual window is a frame of a specified size surrounding a keyword. If a keyword appears several places in the document then several windows are created. Each keyphrase is stemmed to remove duplicates of the same word or phrase by finding their common root. If a duplicate is found then the frequencies are summed up and the duplicate removed. Finally, those candidates above a specified frequency threshold (see Figure 2) are kept and stored in a *document feature vector (dfv)* for that document.

Step 2.4: Cluster search results

In order to identify (discriminate) different domains within the documents found for each entity, clustering techniques are used. At this stage of the process the ontology entities are treated as ordinary terms and can consequently be part of many different domains, which

is a typical problem for IR systems in general. Clustering allows finding different domains. Currently the Lingo [12] and the K-means [10] algorithms are implemented, however in the experiment described in section 4 Lingo was used. The input to the clustering algorithm can either be the content (snippet or full text) of each page found in Step 2.2 or the extracted keywords found in Step 2.3. The result of this step is a set of clusters for each entity. In addition, for each cluster a *cluster feature vector (cfv)* is associated. The associated *cfv* is a combination of all relevant *dfvs* extracted from each of the pages (Step 2.3) of the cluster.

Step 3: Identify and construct

This step constitutes two sub-steps where the aim is to identify the most relevant clusters w.r.t. the ontology and create the final feature vectors.

Step 3.1: Identify domain relevant clusters

A problem at this stage is to identify the correct domain, that is, the most relevant clusters found in Step 2.4 w.r.t. the ontology. Therefore, we compute the similarity between the *cfvs* of an entity with the *cfvs* of its neighboring entities. If a neighboring entity already has been assigned an *fv* (Step 3.2) then that *fv* is used instead of computing the similarity with all the candidate *cfvs* of the entity found in Step 2.4. Recall Step 1, where we argued the importance of finding the most representative entities of the ontology. This ranked list of entities found in Step 1 is important in this step of identifying the most relevant clusters of the domain. Populating the most central entities and then use those *fvs* when selecting the domain for new entities has several advantages. Firstly, those selected clusters have higher relevance to the ontology since the selected cluster of the neighboring entities also is considered in the process. Alternatively, all the clusters of an entity are compared with all the neighboring clusters independent of the selected clusters of the neighbors, which will always be the case of the first entity to be processed. Secondly, the algorithm becomes more efficient while it does not have to calculate the similarity with all the clusters of an already processed entity but only the assigned *fv*.

Commonality (i.e. high similarity) here identifies the document sets (clusters) being relevant to the domain of our interest. The hypothesis is that individual clusters having high similarity across ontology entities are with high probability of the same domain. This hypothesis is backed up with observed patterns of collocated terms within the same domain, and consequently different domains will have different collocation pattern of terms. However, the similarity of clusters depends a lot on the quality of the ontology, especially how much the different entities overlap. The process starts with the first entity of the ranked list entities created in Step 1. The result of this step is a Domain Relevance Measure score for each cluster of an entity. The relations of each entity are given different weighting according to Definition 5.

Definition 5: *Domain Relevance Measure (DRM)*

³ Yahoo! Inc, <http://www.yahoo.com>

⁴ Google Inc., <http://www.google.com>

⁵ Nutch, <http://lucene.apache.org/nutch>

⁶ Stanford POS Tagger, <http://nlp.stanford.edu/software/tagger.shtml>

⁷ FastTag, <http://www.markwatson.com/opensource/>

Let $S = \{S_1, S_2, S_3, S_4, S_5\} = \{\text{subclasses}[e], \text{superclasses}[e], \text{individuals}[e], \text{siblings}[e], \text{relations}[e]\}$, $c_i \in \{\text{clusters}[e]\}$, and $c_k \in \{\text{clusters}[S_j]\}$

$$DRM(e, c_i) = \sum_{j=1}^5 \frac{1}{n_j} \sum_{k=1}^{n_j} w_j S_j \text{sim}(c_i, c_k) \quad (5)$$

where $DRM(e, c_i)$ is the Domain Relevance Measure for entity e and cluster c_i of e . w_j is a weight factor set to a default value of 1, and S_j is either 1 if S_j is true or 0 if S_j is false. Further, n_j is the number of clusters of each neighboring entity defined in S .

Note, that if a neighboring entity already has an fv assigned (Step 3.2) then that fv will be used in the calculation of the DRM for that entity instead of calculating the similarity with all its clusters.

Step 3.2: Construct feature vector

The cluster with the highest DRM score, calculated in Step 3.1, is selected for each entity. The step of creating the final fv for the selected cluster can either be based on the already created $clfv$ of that cluster (Step 2.4) or a deeper analysis of the documents of the selected cluster can be done.

3.3 Feature Vector Construction

Example

In this section, a small example is presented to illustrate the steps of the Feature Vector Construction algorithm described in Section 3.2.

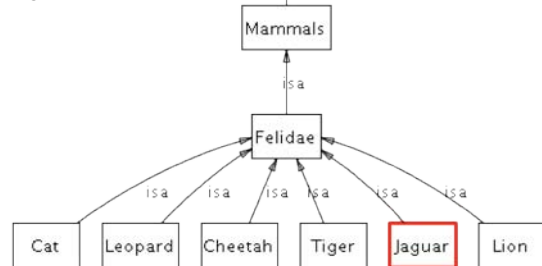


Figure 3. A small fragment of the Animals⁸ ontology, where the Jaguar entity is highlighted and used in this example.

Step 1: Rank entities

The example ontology is presented in Figure 3. A ranked list with default weighting values is shown in Table 1. Note, that the Animals ontology contains 51 entities and consequently only some of them are shown in Table 1 to save space.

Table 1. Ranking of the entities found in the Animals ontology presented in Figure 3.

Ranking	Entity	EDM	EBM	ECS
1.	Animals	1,00	0,58	0,79
2.	Mammals	0,97	0,58	0,78
4.	Felidae	0,47	0,42	0,44
26.	Jaguar	0,23	0,00	0,11
28.	Lion	0,23	0,00	0,11
29.	Tiger	0,23	0,00	0,11

⁸ Animals ontology, <http://nlp.shef.ac.uk/abraxas/ontologies/animals.owl>

51.	Monkey	0,09	0,00	0,05
-----	--------	------	------	------

Step 2.1: Create entity query

For the entity Jaguar, seen in Figure 3, the search query will be as follows:

felidae jaguar

In this case comprehensive search task is used to select which neighboring entities to include. However, siblings are not include for the expanded query because the number of siblings can be many and instead add noise to the query. Consequently only the super-class is included which is Felidae.

Step 2.2: Entity based search

A search based on the query created in Step 2.1 is performed. The three top ranked document by Yahoo!, as of 7th of March 2009 is shown in Table 2.

Table 2. Top three search results for "felidae jaguar".

1. Jaguar - Wikipedia

Article about the jaguar (*Panthera onca*), the New World mammal of the felidae family and one of four "big cats" in the Panthera genus.

<http://en.wikipedia.org/wiki/Jaguar>

2. Lioncrusher's Domain -- Jaguar (*Panthera onca*) facts and pictures

Lioncrusher's Domain > felidae > jaguar ... Comparison of jaguar spots, left, to Leopard spots, right. Many people get the jaguar and leopard confused. ...

<http://www.lioncrusher.com/animal.asp?animal=53>

3. Felidae - Wikipedia, the free encyclopedia

The late Miocene radiation of modern felidae: a genetic assessment" ... Lion (*P. leo*) · jaguar (*P. onca*) · Leopard (*P. pardus*) · Tiger (*P. tigris*) Uncia ...

<http://en.wikipedia.org/wiki/Felidae>

Step 2.3: Contextual keyword extraction

For illustration purposes a small text fragment is used to illustrate contextual keyword extraction process. The contextual window size was 200 characters.

Table 3. A text fragment from the top search result (Table 2) is shown at the top and a set of corresponding extracted keywords for the whole document is seen at the bottom.

Text fragment of the top search result (Table 2)

"The jaguar, *Panthera onca*, is a New World feline and one of four "big cats" in the Panthera genus, along with the tiger, lion, and leopard of the Old World. It is the only Panthera found in the New World. The jaguar is the third-largest feline after the tiger and the lion, and on average the largest and most powerful feline..."

Extracted keywords from the whole page

animal (22), cat (44), civet (18), genet (17), habitat (16), jaguar (166), jaguar panthera onca (5), mexico (14), mongoose (34), panthera (14), panthera onca (12), population (17), prey (22), range (31), species (43), state (15)

Step 2.4: Cluster search results

We used the Lingo clustering algorithm. 30 documents are used in this case as input to the clustering algorithm. The full documents of the top 30 documents presented by Yahoo! were used. The result was three clusters as shown in Table 4.

Table 4. *fvs* for clusters found for the entity "jaguar".

<p>Cluster#1= {animal (111), august (25), big cat (6), cat (178), cat rescue (5), day (7), facts (7), felis (10), felidae (8), fish (6), full text (15), full text panthera onca (3), genus (13), habitat (59), jaguar (813), jaguar panther panthera onca (2), jaguar panthera onca (26), johnson (17), leopardus (10), lineage (7), lynx (9), male (8), mate (6), mexico (40), mongoose (28), nowell (13), onca felis onca (2), other big cats (3), page (6), panthera (36), panthera onca (53), panthera onca felis (2), population (17), prey (104), prionailurus (6), range (120), retrieved june (4), species (214), state (40), territory (7), travel (18), wild (6), world encyclopedia (4)}</p> <p>Cluster#2= {animal (24), caracal (8), cat (97), civet (12), felis (9), felid (11), felidae (12), felinae (9), genus (14), habitat (16), jaguar (164), jaguar panthera onca (5), leopard (8), leopardus (11), lineage (9), lynx (15), mexico (14), mongoose (50), palm (9), panthera (21), panthera onca (12), population (17), prey (22), prionailurus (8), range (31), species (43), state (14)}</p> <p>Cluster#3= {britannica concise encyclopedia (3), encyclopædia britannica (7), home library (8), jaguar (72), reserved read (6), rights (9), site (6), state (10), top home (6)}</p>

Step 3.1: Identify domain relevant clusters

By calculating the similarity with the clusters of the neighboring entities of *Jaguar*, which are *Felidae* (super-class) we can identify the relevant cluster to this domain. In this case *Cluster#2* had the highest similarity (see Table 5) with a DRM score of 0,267. This cluster is therefore selected as the candidate cluster for the construction of the feature vector to be done in the next step.

Table 5. Cluster DRM for the entity "jaguar".

Cluster #	DRM
2	0,267
1	0,119
3	0,000

Step 3.2: Construct feature vector

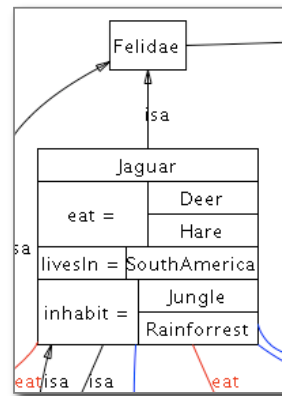
The last step for the *Jaguar* entity is to create the final entity feature vector, which in this example will be the same as the *clfv* for *Cluster#2* as seen in Table 6. At this stage we could do a more thoroughly analysis of the cluster documents to improve the quality of the feature vector even further.

Table 6. The final *fv* created for the *Jaguar* entity.

<p>Jaguar= {animal (24), caracal (8), cat (97), civet (12), felis (9), felid (11), felidae (12), felinae (9), genus (14), habitat (16), jaguar (164), jaguar panthera onca (5), leopard (8), leopardus (11), lineage (9), lynx (15), mexico (14), mongoose (50), palm (9), panthera (21), panthera onca (12), population (17), prey (22), prionailurus (8), range (31), species (43), state (14)}</p>
--

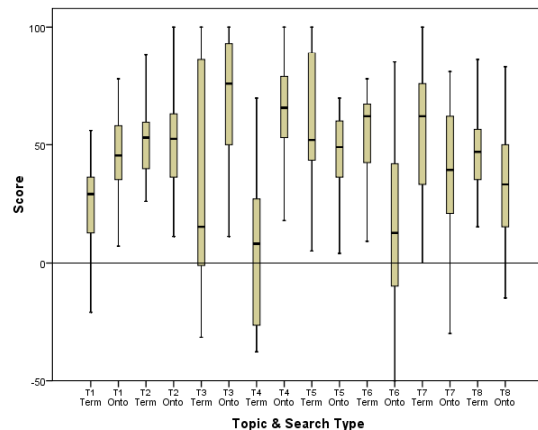
4. EXPERIMENT

In this section, we present an experiment performed in the first half of 2008. We describe the prototype, the design and in the next subsection we present the results.

**Figure 4.** A fragment of the extended Animals ontology showing the *Jaguar* entity with added object properties.**4.1 Experiment Settings**

WebOdIR⁹ was implemented in Java and run on a Tomcat server. The prototype used the Yahoo! Web Search API¹⁰ as the backend search engine.

The participants in our experiment were mainly 4th year students at the Norwegian University of Science and Technology (NTNU). There were 21 subjects that participated; they were offered payment for used time after full completion of the experiment.

**Figure 5.** A box plot graph showing the result score for all the topics for both the concept- and keyword-based search. *T1* to *T8* are the topics one to eight respectively found in [17]. *Term* is the keyword-based search and *Onto* is the concept-based search for each topic. The score is in the range from -50 to 100.

The experiment consisted of two steps. The first step included formulating search queries for both WebOdIR and the baseline (Yahoo! Web Search). Four domains with two topics descriptions for each domain were

⁹ WebOdIR prototype (08.03.2009), <http://129.241.110.220>

¹⁰ Yahoo! Developer Network, <http://developer.yahoo.com>

presented. They had to formulate in total 16 queries, eight to be submitted to WebOdir and eight to the baseline. The participants were also divided into two groups to test how different granularities of the ontologies would influence on the search results [17] (Figure 3 vs. 4). After finishing the practical part they had to perform the second step, which was to complete a questionnaire of 29 questions.

In this first experiment we choose to use the snippets from Yahoo! instead of the full text documents. The main reason for this was performance, cause for each search the top 100 documents was evaluated and downloading the top 100 pages took in average more than two minutes to complete. The snippets from Yahoo! Web Search had an average length of 142 characters.

Table 7. Average performance scores and correlation table.

	Average scores				Correlation			
	Wine	Travel	Animals	Autos	Term-search	Onto-search	Familiarity	Helpfulness
Term-based search	39,0	21,4	55,3	53,0	1			
Ontology-driven search	46,9	68,3	29,5	33,4	-0,999	1		
Familiarity w/domain*	2,38	2,62	2,74	2,71	0,433	-0,395	1	
Helpfulness of ontology*	3,45	3,67	3,26	2,88	-0,840	0,818	-0,480	1

Remark: * - measured with a Likert 5 point scale, where 1 - lowest value, and 5 - highest.

The results from the Yahoo! Web Search was selected as the baseline for our comparisons since it was used as the backend search engine. Ideally Text Retrieval Conference (TREC)¹¹ would be used but we experienced the same problems as d'Acquin et al. [7] in finding good ontologies that covered TREC.

The participants had to do a qualitative perceived relevancy of the top 10 results. We adopted the query scoring and calculation strategy presented by Brasethvik [4]. The participants needed to mark each of top 10 retrieved documents according to perceived relevance. The relevance score for each query has been calculated using the following equation:

$$Score_q = \frac{1}{2} \sum_{i=1}^{10} P_{D_i} \times P_{P_i} \quad (6)$$

where P_{D_i} is an individual score for document D_i , and P_{P_i} - the weighting factor for position P_i . The score for a document is as follows: -1 for *trash*; 0 for *irrelevant* or *duplicate*; 1 - *related*; and 2 - *good document*. Document ranking positions have weights as follows: 1st - 20; 2nd - 15; 3rd - 13; 4th - 11; 5th - 9; 6th & 7th - 8; 8th & 9th - 6; 10th - 4. Consequently, the final score falls into a range [-50, 100].

Table 8. Average relevance scores versus ontology version.

	Ontology ver. 1	Ontology ver. 2	Diff. (%)
Animals	19.4	38.0	96.6%
Autos	32.9	33.7	2.2%
Travel	71.8	65.2	-9.1%
Wine&Food	42.9	51.8	20.6%
Overall	42.1	46.6	10.7%

The relevance score substitutes a conventional precision metric. We have decided to focus on precision instead of recall since we aimed at improving Web search results, where precision (i.e. relevant documents at top positions) is more important.

4.2 Results

Figure 5 summarizes the main results of the experiment. The concept-based search performs in general better than term-based search for the Wine (T1&T2) and Travel (T3&T4) ontologies, but worse for the Animals (T5&T6) and Autos (T7&T8) ontologies. One of the reasons for concept-based search performing worse for the two last domains is the quality of those ontologies [17]. Furthermore, the Wine and Travel ontologies had more familiar terms used to denote concepts than the Animals and Autos ontologies did. Despite of this, most users expressed in the survey that they in average were more familiar with the topic descriptions of Animals and Autos than Wine and Travel (see Table 7), and consequently, they found the ontologies less helpful regarding the topics for Animals and Autos.

Table 9. Ontology and feature vector characteristics.

Domain	Ontology version	Ontology characteristics			Feature vectors' characteristics	
		# of concepts	# of instances	# of properties	average length	average cosine similarity
Wine and food	Onto 1	82	155	14	36,66	0,92
	Onto 2	83	157	17	38,38	
Travel	Onto 1	34	14	10	34,67	0,92
	Onto 2	34	29	10	37,26	
Animals	Onto 1	51	0	2	33,04	0,78
	Onto 2	63	15	8	36,12	
Autos	Onto 1	90	321	16	33,27	0,87
	Onto 2	91	328	16	33,65	

All four ontologies were modified by adding instances (all ontologies), specifying additional object properties (Travel, Animal, and Wine ontologies) and introducing equivalent classes (Animal and Auto ontologies) (Table 9). This difference in granularity affected the quality of the feature vectors (see Table 9 and 10).

Table 10. Listing of the keyphrases of the feature vectors for the entity Jaguar from the Animal ontology. Version 1 indicates the original ontology (Figure 3) while Version 2 indicates the extended version (Figure 4).

¹¹ Text Retrieval Conference (TREC), <http://trec.nist.gov>

Animals Ontology - Jaguar	
Version 1	Version 2
animal	animal
animal jaguar	animal jaguar
animal jaguar conservation	animal jaguar conservation
animal jaguar farms	animal jaguar farms
animal jaguar population	animal jaguar population
animal muscular stocky member	conservation
cat	conservation measures
conservation	farms
conservation measures	hunting
farms	photos
found	taken
hunting	wild animal jaguar
Information about	wild cats often
jaguar	you can see
jaguar animal muscular	
photos	
rain forests	
taken	
wild animal jaguar	
wild cat	
wild cats often	
you can see	

The difference in relevance scores for the original ontologies versus the modified ones; we found an improvement in the mean score that equals 10.6% (see Table 8). This indicates that in general a more advanced ontology in the sense of having more relations, properties and individuals does perform better than a similar simpler ontology. A reason for this can be that for the more advanced ontologies more knowledge is available in the process of creating the entity *f/s* and hence will contain less noise compared to those of a simpler ontology.

5. CONCLUSION

In this paper, we have presented an approach that utilizes ontologies to enhance the effectiveness of large-scale search systems for the Web. We have described the overall architecture and explained how such systems can potentially be improved by a pragmatic usage of ontology semantics together with the specific terminology used in the actual text corpora, i.e. the Web. This is done by construction of a feature vector for each of the ontology entities. The feature vector typically contains terms that are associated with the concepts reflected by the document collection.

An experiment conducted in 2008 was presented. A prototype was developed and real users evaluated its performance. Analysis of the experiment showed that the approach performed well in some domains but worse in others. In this experiment ontologies with different granularity where used, we have shown how this affect the quality of the feature vectors and hence the search quality.

In future work we will look into alternative methods for post-processing of the retrieved documents utilizing the semantic relations in the ontology for better ranking and navigation. Furthermore, one of the major future researches lies in how to better tailor feature vector construction to various search tasks (i.e., fact-finding, explorative and comprehensive) and to research different techniques in order to reduce sensitivity of the approach to quality of ontology.

6. ACKNOWLEDGMENTS

This research work is funded by the Integrated Information Platform for reservoir and subsea

production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30. In addition, we would like to thank Jon Atle Gulla (NTNU), Per Gunnar Auran (Yahoo!), and Robert Engels (ESIS) for their support and help.

7. REFERENCES

- [1] Alani, H., Brewster, C., Shadbolt, N.: Ranking ontologies with AKTiveRank. In ISWC 2006, LNCS 4273, 1-15. Springer-Verlag (2006)
- [2] Bergamaschi, S., Bouquet, P., Giazomuzzi, D., Guerra, F., Po, L., Vincini, M.: An Incremental Method for the Lexical Annotation of Domain Ontologies. *Int. J. on Semantic Web and Information Systems* 3(3) (2007) 57-80
- [3] Braga, R.M.M., Werner, C.M.L., Mattoso, M.: Using Ontologies for Domain Information Retrieval. *Proceedings of DEXA'00. IEEE Computer Society* (2000) 836-840
- [4] Brasethvik, T.: Conceptual modelling for domain specific document description and retrieval - An approach to semantic document modelling. PhD thesis, IDI,NTNU, Trondheim (2004) 257
- [5] Bry, F., Koch, C., Furche, T., Schaffert, S., Badea, L., Berger, S.: Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages. *Int. J. on Semantic Web and Information Systems* 1(2) (2005) 1-21
- [6] Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. *IEEE TKDE* 19(2) (2007) 261-272
- [7] d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Toward a New Generation of Semantic Web Applications. *IEEE Intelligent Systems* 23 (2008) 20-28
- [8] Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, Vol. 1. Cambridge University Press (1995) 9-27
- [9] Lytras, M.D., Sicilia, M-A.: Where is the value in metadata? *Int. J. of Metadata, Semantics and Ontologies* 2(4) (2007) 235-241
- [10] Manning, C.D., Schütze, H.: *Foundations of Statistical Natural Language Processing*. The MIT Press (1999)
- [11] Nagypal, G.: Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies. *OTM Workshops 2005, LNCS 3762*, Springer-Verlag, (2005) 780-789
- [12] Osinski, S., Weiss, D.: A Concept-Driven Algorithm for Clustering Search Results. *IEEE Intelligent Systems*, Vol. 20 (2005) 48-54
- [13] Ozcan, R., Aslangdogan, Y.A.: Concept Based Information Access Using Ontologies and Latent

Semantic Analysis. Technical Report CSE-2004-8. University of Texas at Arlington (2004) 16

[14] Panagis, Y., Sakkopoulos, E., Garofalakis, J. and Tsakalidis, A. Optimisation mechanism for web search results using topic knowledge. *Int. J. Knowledge and Learning* 2(1/2) (2006) 140–153

[15] Solskinnsbakk, G., Gulla, J.: Ontological Profiles as Semantic Domain Representations. *Natural Language and Information Systems* (2008) 67-78

[16] Solskinnsbakk, G., Gulla, J.: Ontological Profiles in Enterprise Search. *Knowledge Engineering: Practice and Patterns* (2008) 302-317

[17] Strasunskas, D., Tomassen, S.L.: Empirical Insights on a Value of Ontology Quality in Ontology-driven Web Search. In: Meersman, R., Tari, Z. (eds.): OTM 2008, Part II, LNCS.1319-1337. Springer-Verlag (2008)

[18] Suomela, S., Kekalainen, J.: Ontology as a search-tool: A study of real user's query formulation with and without conceptual support. In: Proceedings of ECIR'2005. LNCS 3408, Springer-Verlag (2005) 315-329

[19] Yang, H-C. A method for automatic construction of learning contents in semantic web by a text mining

approach. *Int. J. Knowledge and Learning* 2(1/2), (2006) 89–105

APPENDIX

A.1 POS tagging rules

Tagging rules of length 1

NN, NNS, NNP, NNPS, JJ, VBG

Tagging rules of length 2

NN-NN, NN-NNP, NN-NNS, JJ-NN, JJ-NNS, JJ-NNP, NNP-NNP, VBN-NNP

Tagging rules of length 3

JJ-JJ-NN, JJ-JJ-NNS, JJ-NN-NN, JJ-NN-NNS, JJ-NNS-NN, JJ-NNS-NNS, NN-JJ-NN, NNP-JJ-NNP, NN-JJ-NNS, NNS-JJ-NN, NNS-JJ-NNS, NN-NN-NN, NN-NN-NNS, NN-NNS-NN, NNS-NN-NN, NNS-NN-NNS, NNS-NNS-NN, NNS-NNS-NNS, NN-IN-NN, NN-IN-NNS, NNS-IN-NN, NNS-IN-NNS

Tagging rules of length 4

JJ-NN-NN-NN, JJ-NN-NN-NNS, NN-VBN-NN-JJ, NN-IN-NN-NN, NN-NNS-NN-NNS, NN-NN-NN-NN

P2: Semantic-Linguistic Feature Vectors for Search: Unsupervised Construction and Experimental Validation

Publication details

Tomassen, S.L. & Strasunskas, D. (2009) Semantic-Linguistic Feature Vectors for Search: Unsupervised Construction and Experimental Validation. In: Gomez-Perez, A., Yu Y. & Ding Y. (eds.) *The Semantic Web*, LNCS 5926, Springer, Heidelberg, pp. 199-215.

Semantic-Linguistic Feature Vectors for Search: Unsupervised Construction and Experimental Validation

Stein L. Tomassen¹, Darijus Strasunskas²

¹ Dept. of Computer and Information Science

² Dept. of Industrial Economics and Technology Management
Norwegian University of Science and Technology, Norway

¹stein.l.tomassen@idi.ntnu.no, ²darijuss@gmail.com

Abstract. In this paper, we elaborate on an approach to construction of semantic-linguistic feature vectors (FV) that are used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by a relevant terminology from Web documents. The value of this approach is twofold. First, it captures relevant semantics from an ontology, and second, it accounts for statistically significant collocations of other terms and phrases in relation to the ontology entities. The contribution of this paper is the FV construction process and its evaluation. Recommendations and lessons learnt are laid down.

1 Introduction

Search is among the most frequent activities on the Web. However, the search activity still requires extra efforts in order to get satisfactory results. One of the reasons is heterogeneous information resources and exponential growth of information. There are many different approaches proposing a solution for this problem. Some approaches are relying on semantic annotations (e.g., [2, 19]) by adding additional metadata; some are enhancing clustering of retrieved documents according to topic (e.g. [13]); some are developing powerful querying languages (e.g. [4]). Therefore, many efforts are devoted to research on improvement of information retrieval (IR) by the help of ontologies that encode domain knowledge (e.g. [5, 17]).

The objective of this paper is to discuss our approach to semantic search that builds on a concept of *feature vector* (FV) and elaborate on the FV construction (FVC) process. The approach is based on pragmatic use of ontologies by relating the concepts (domain semantics) with the actual terminology used in a text corpus, i.e. the Web. We propose to associate every entity (classes and individuals) of the ontologies with a FV to tailor them to the terminology in a text corpus. First, these FVs are created off-line and later used on-line to filter, and hence disambiguate search, and re-rank the search results from the underlying search system. The proposal is based on a non-supervised solution that is applicable to any ontology as long as there is some correlation between the ontology and the text corpus. Moreover, the approach is independent from a collection of relevant documents. Possibility to use a diverse corpus (the Web) is the main advantage of the approach since the approach builds on word sense disambiguation by utilizing the relationships between the entities. Nevertheless, the FV quality will be highly depended on both the quality of the ontology and the correlation of terminologies in the ontology and the text collection.

In [15], we focused on FVs used to disambiguate search that was evaluated with real users. While in [18], the FVC algorithm used in Strasunskas and Tomassen [15] was presented. Therefore, in this paper we focus on the aspects of the components of FV construction algorithm that

affect the feature vector quality. Furthermore, in the evaluation we analyse the effect of alternative techniques on the FVs.

Moreover, many approaches build on similar artefacts as our FVs, although they target various application areas (e.g., ontology alignment, ontology mapping, semantic search, ontological filtering), cf. [7, 9, 14, 16]. Despite they are differently built, this paper provides useful insights on how the process of FVC can be evaluated and the FV quality assessed.

This paper is organized as follows. In section 2, related work is discussed. In section 3, the algorithm of how the FVs are constructed and a small example of the process are presented. In Section 4, we present the conducted experiments and explain the evaluation. Then in section 5, the results will be analyzed. Finally, in section 6, we conclude this paper.

2 Related Work

The focus of this paper is the construction of feature vectors (FV). Therefore, scope of related work synopsis provided here is limited correspondingly. In general, FVs can be classified in three groups, numerical, textual, and a mixture of both. Numerical FVs are typically used in machine learning (e.g. [10]) and are not relevant here, which neither is the case for approaches using mixed FVs. Textual FVs on the other hand, are typically based on a lexical resource like WordNet (e.g. [9]) or extracted from a set of documents (e.g. [1, 14, 16, 20]). The latter form of FVs is most relevant and will be reviewed in more details.

There are approaches that depend on highly relevant document collections (e.g. [14, 16]) as distinct from our approach. Approaches that are more interesting are based on topic signatures. A topic signature is a list of topically related words [1]. There are many topic signature approaches (e.g., [1, 20]. Zhou et al. [20] propose a Topic Signature Language Model that is used to perform semantic smoothing to increase the retrieval performance. They create topic signatures for each concept defined in domain specific ontology using a highly relevant document collection. The topic signature terms are found by collocation. They assume that the concepts are unique and consequently circumvent the problem of word disambiguation. For general domains where no ontology exists, they propose to use multiword expressions as topic signatures. The multiword expressions contains context in nature and are consequently mostly unambiguous.

While Agirre et al. [1] propose enriching WordNet with topic signatures using the Web. A concept in WordNet can contain several senses. Nevertheless, for each sense a set of cue-words (hyponyms, hypernyms, etc.) is used to create a highly specific query that is submitted to the search engine. The top 100 documents are retrieved and keywords are extracted. They experienced formulating the queries being the weakest point of their approach. The quality of the queries highly affected the quality of the retrieved documents. In contrast to our approach that is not depended on a high quality query but uses clustering and domain identification, based on neighbouring entities, to find relevant documents from a set of diverse documents.

3 Feature Vector Construction

Every ontology entity has an associated feature vector with a set of relevant terms extracted from the text corpora. An ontology entity can be either a class or an individual. In this approach, we use the term *entity* instead of *concept* because a concept is often a synonym for a class when it comes to ontologies. Our approach associates feature vectors to both classes and individuals that hereinafter are referred to as entities. In this section, we will describe the process of how these FVs are constructed, but first a definition of a feature vector is provided. At the end of this section, an example of the construction process is presented.

3.1 Definition of a Feature Vector

The development of the approach is inspired by a linguistics method for describing the meaning of objects - the semiotic triangle [11]. In our approach, a feature vector "connects" a concept (entity) to a document collection, i.e., the FV is tailored to the specific terminology used in a particular collection of the documents. FVs are built considering both semantics encoded in an ontology and a dominant lexical terminology surrounding the entities in a text corpus. Therefore, a FV constitutes a rich representation of the entities and is related to actual terminology used in the text corpus. Correspondingly, a FV of an entity e is represented as a two-tuple (see Definition 1):

Definition 1: *Feature Vector (FV)*

$$FV_e = \langle S_e, L_e \rangle \mid S_e \in O_d, L_e \in D_d \quad (1)$$

$$S_e = (e_i, DR_{e_i})$$

$$DR_{e_i} = Parents_{e_i} \cup Children_{e_i} \cup Others_{e_i} = \{e_i, e_k\} \subseteq E \times E$$

$$L_{e_i} = collocated(S_{e_i}, L_{e_{Dd}})$$

where S_e is a semantic enrichment part of FV_e that represents a set of neighbourhood entities and properties in an ontology O of a domain d . L_e is a linguistic enrichment of a entity that is a set of terms (from document collection D of a particular domain d) with a significant proximity to an entity and its semantic neighbourhood.

3.2 Feature Vector Construction

The Feature Vector Construction (FVC) process is visualized in Figure 1. The algorithm constitutes two phases (main steps). The first phase aims to extract and group candidate terms being potentially relevant to each entity. However, the candidate terms are not necessarily relevant to the domain defined by the ontology (terms can be ambiguous). Consequently, the aim of the last phase is to identify those groups of candidate terms being relevant to the entities w.r.t. the ontology. Finally, an FV for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding FVs, which consist of terms associated to both the entities and the domain terminology (Eq. 1).

The FVC algorithm is designed to be flexible in the sense that it can be tailored to the intended usage of the FVs as well as the different quality of the ontologies. Consequently, the algorithm provides several options at each step. The effect of some of these options is evaluated in section 4 and 5, while detail description follows below.

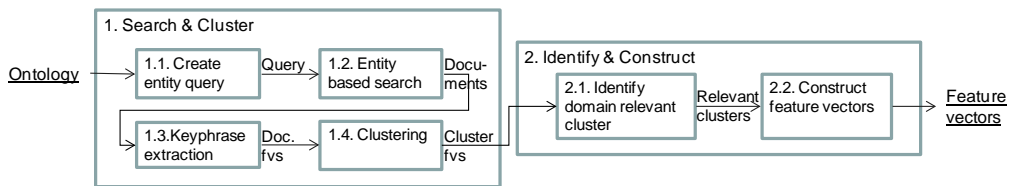


Fig. 1. The Feature Vector construction process

Step 1: Search and cluster

This step constitutes four sub-steps where the aim is to extract candidate terms that are relevant to each entity. The candidate terms are grouped and then, in Step 2, further processed to identify which of the candidate groups being most relevant to the domain of interest defined by the ontology.

Step 1.1: Compose entity query

In this step, a search query is prepared for each entity while the actual search is performed in Step 1.2. The query is based on the entity label with an option to include relevant neighbouring entities and/or keyword(s) (more of this Section 4.2). Here we aim at creating a query that reflects on the ontology by considering closest neighbours of a particular entity.

A *parent* of a class is defined to be its super class, while a parent of an individual is the class the individual being an instance of. A *child* of a class is defined to be its sub class or individual, the latter if it does not have a sub class. An individual does not have a child. Finally, *other* neighbouring entities are any other object property defined in OWL. The motivation behind expanding the initial query with neighbouring entities is to create a query that reflects both the ontology and the relationship of each entity to other neighbouring entities.

Larger ontologies tend to include several minor domains. By experimentation we found that for diverse ontologies, like the Wine¹ ontology that also imports the Food² ontology, it can be beneficial to add keyword(s) that represents the overall subject domain. The result of using keyword(s) is less unique and more homogeneous FVs while omitting keywords would create FVs that are more unique and more true to the local variances in the ontology.

Step 1.2: Entity based search

The query for each entity created in Step 1.1 is used to retrieve candidate documents for each entity. Any search engine can be used in this step. Currently, Yahoo! and Google (for searching in Web documents) and Nutch³ (for searching in local documents) are supported. In the experiments described in Section 4 Yahoo! is used. The retrieval session is keyword-based.

Step 1.3: Contextual key-phrase extraction

For each document a set of key-phrases and keywords is extracted, hereinafter referred to as key-phrases. First, a part of speech (POS) tagger is used to tag the retrieved documents (snippet or full text). In the experiments described in Section 4 we have selected to use FastTag⁴, because it is fast and by experiments found it to perform adequate on Web documents and snippets with diverse quality.

Then a set of tagging rules (39 rules), inspired by Justeson and Katz [8], is applied. Based on these rules a set of candidate noun key-phrases are extracted. However, only those key-phrases within what we call a contextual window are extracted. A contextual window is a frame of a specified size surrounding a keyword (in the experiments described in section 4 a window of size 50 is used). If a keyword appears several places in the document then more windows are created. Each key-phrase is stemmed to remove duplicates by finding their common root. If a duplicate is found then the frequencies are summed up and the duplicate removed. Finally, those candidate key-phrases above a specified frequency threshold (dependent on the document length) are kept and stored in the *document feature vector* (DFV) of the corresponding documents.

Step 1.4: Cluster search results

In order to identify (discriminate) different subject domains within the documents found for each entity, clustering techniques are used. Recall that the retrieval session is keyword-based (Step 1.2) consequently the terms (entities) can be part of many different domains. Clustering allows us to find these different domains. Currently the Lingo [12] algorithm is used since it performs well for both snippets and full-text documents. The result of this step is a set of clusters for each entity. In addition, for each cluster a *cluster feature vector* (CLFV) is created. A CLFV is a combination of all the DFVs of a cluster. In the following step, we deal with selecting the relevant cluster w.r.t. the domain of interest.

¹ Wine, <http://www.w3.org/2001/sw/WebOnt/guide-src/wine.owl>

² Food, <http://www.w3.org/2001/sw/WebOnt/guide-src/food.owl>

³ Nutch, <http://lucene.apache.org/nutch>

⁴ FastTag, <http://www.markwatson.com/opensource/>

Step 2: Identify and construct

This step is constituted of two sub-steps and aims at identifying the most relevant clusters w.r.t. the ontology and create the final feature vectors.

Step 2.1: Identify domain relevant clusters

A problem at this stage is to identify the correct subject domain, that is, the most relevant clusters found in Step 1.4 w.r.t. the ontology. Therefore, we compute the similarity between the cluster feature vectors of an entity with the CLFVs of the selected neighbouring entities. In order to find the most prominent cluster, an entity must have at least one neighbour otherwise this check would fail. The neighbouring entities are grouped according to their relation type, as in Step 1.1, i.e., *parents*, *children*, and *other* entities.

Commonality (i.e. high similarity) here identifies the document sets (clusters) being relevant to the domain of our interest. The hypothesis is that individual clusters having high similarity with neighbouring entities are with high probability of the same domain defined by the ontology. This hypothesis is backed up with observed patterns of collocated terms within the same domain, and consequently different domains will have different collocation pattern of terms. However, the similarity of clusters depends a lot on the quality of the ontology, especially on semantic distance between the different entities.

The result of this step is a Domain Relevance Measure score for each cluster of an entity. The relations of each entity are given different weighting according to Definition 2.

Definition 2: Domain Relevance Measure (DRM)

Let $S = \{S_1, S_2, S_3\} = \{\text{parents}[e], \text{children}[e], \text{other}[e]\}$, $c_i \in \{\text{clusters}[e]\}$, and $c_k \in \{\text{clusters}[S_j]\}$

$$DRM(e, c_i) = \sum_{j=1}^3 \frac{1}{n_j} \sum_{k=1}^{n_j} w_j S_j \text{sim}(c_i, c_k) \quad (2)$$

where $DRM(e, c_i)$ is the Domain Relevance Measure for entity e and cluster c_i of e . w_j is a weight factor set to a default value of 1, and S_j is either 1 if S_j is true or 0 if S_j is false. Further, n_j is the number of clusters of each neighbouring entity defined in S .

Step 2.2: Construct feature vector

The cluster with the highest DRM score, calculated in Step 2.1, is selected for each entity. The step of creating the final FV for the selected cluster can either be based on the already created CLFV of that cluster (Step 1.4) or a deeper analysis of the documents of the selected cluster can be done. In the experiments described in section 4, the CLFVs were used.

3.3 Feature Vector Construction Example

In this section, a small example is presented to illustrate the steps of the Feature Vector Construction algorithm described in Section 3.2.

Step 1.1: Create entity query

In order to better illustrate the purpose of the clustering (step 1.4) and the identification of the domain relevant clusters in step 2.1, the illustrative query for the entity **Jaguar**, seen in Figure 3, is: <jaguar>

Step 1.2: Entity based search

The query created in Step 1.1 is submitted to Yahoo! Search and the three top ranked documents (of 30 used in this example), as of 18th of April 2009, are shown in Table 1. Not surprisingly was **Jaguar** the car brand most popular for the moment (23 of 30 top ranked), then panther (5/30), perfume (1/30), and vodka (1/30).

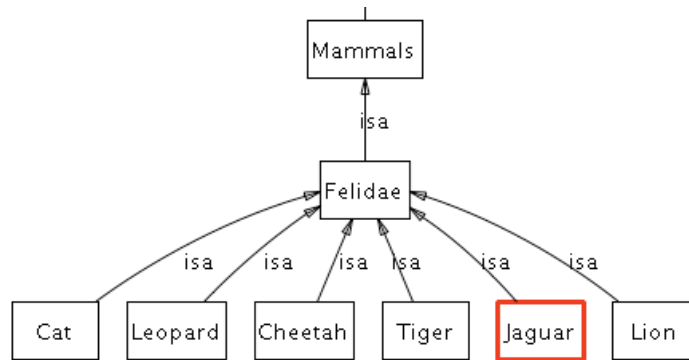


Fig. 2. A small fragment of the Animals⁵ ontology, where the Jaguar entity is highlighted and used in this example

Table 1. Top three search results for jaguar.

<p>1. Jaguar Official site of Jaguar featuring new models and local dealer information. http://www.jaguar.com</p> <p>2. Jaguar US – Home Jaguar USA official website ... Build Your XK. Find Your XK. Locate a Dealer. Build Your Jaguar. Find Your Jaguar. Request Brochure ... http://www.jaguarusa.com</p> <p>3. Jaguar - Wikipedia The jaguar, Panthera onca, is a big cat, a feline in the Panthera genus. It is the only Panthera found in the Americas. The jaguar is the third-largest feline after the tiger and the lion, and the largest and most powerful... http://en.wikipedia.org/wiki/Jaguar</p> <p>...</p>

Table 2. A text fragment from the third search result (Table 1) is shown at the top and a set of corresponding extracted key-phrases for the whole document is seen at the bottom

Text fragment of the third search result (Table 1)
"The jaguar, Panthera onca, is a big cat, a feline in the Panthera genus. It is the only Panthera species found in the Americas. The jaguar is the third-largest feline after the tiger and the lion, and the largest and most powerful feline in the Western Hemisphere...."
Extracted key-phrases from the whole page
cat (17), culture (11), habitat (13), jaguar (136), panthera (11), population (11), prey (19), range (20), species (27), state (11)

Step 1.3: Contextual key-phrase extraction

For illustration purposes, only a small text fragment is shown in Table 2 to illustrate the contextual key-phrase extraction process. The contextual window was of size 50. Typical noise in the documents, like menus, is removed. For instance, Wikipedia documents got **start content** and **end content** tags, which are utilized, and hence only the text between these tags is processed.

Step 1.4: Cluster search results

We used the Lingo clustering algorithm. The full text documents were used. Four clusters were created for the jaguar entity as shown in Table 3.

Table 3. FVs for clusters found for the jaguar entity

Cluster#1 = {advice car (4), auto insurance (4), auto show (2), calculators true cost (1), car (94), chevrolet (4), compact awd sport sedan (2), company (13), detailed jaguar (6), drivetrain engine (2), econ msrp (2),
--

⁵ Animals, <http://nlp.shef.ac.uk/abraxas/ontologies/animals.owl>

engine (12), engine trans (2), flagship 4-door sedan line (2), ford (10), information jaguar (2), information pictures (6), invoice (14), invoice price (4), jaguar (197), land rover (7), line (3), low dealer price (4), market value (2), midsize sport sedan (2), model (23), model name (6), motor (7), motor company (6), msrp (16), price (8), quotes inside line (1), review (17), saloon (8), search sitemap company privacy (1), sedan (17), select (4), series jaguar (5), sports (8), stars (14), style (5), system premium sound system (1), terms (1), tips advice (3), trans fuel (2), truck (29), trucks tips advice (1), xj-series (3), yahoo autos (8), zip (4)}

Cluster#2={accolades (2), conditions (2), contact (2), dealer (1), disclaimer international sites (2), features (2), gallery (2), gtr company (2), international sites faq (2), jaguar (2), ownership quality highlights (2), pre-owned (2), privacy policy (2), profile site (2), request brochure (1), site (2), sites faq gtr (2), site map (2), specs (2), terms (2)}

Cluster#3={cat (26), culture (11), habitat (13), jaguar (164), panthera (11), population (11), prey (19), range (20), species (27), state (11)}

Cluster#4={accessories (5), blue grass (1), blvd louisville (2), brake (7), car (10), careers (1), contact info links (1), deal (1), department (1), exterior jaguar (3), fax (1), genuine (10), genuine parts order (1), inventory (1), inventory pre-owned inventory (1), jaguar (204), jaguar blue (1), jaguar brake (5), jaguar fuel (5), jaguar jaguar (71), land rover (2), news (2), order parts service (1), part (24), parts catalogaccessories catalogjaguar (1), part number (13), parts service schedule (1), phone (1), pre-owned (1), pre-owned inventory events (1), rotor part number (2), rover jaguar (1), saab land (1), service (1), service contact (1), service schedule service contact (1), serviceservice (1), shop jaguar (3), specials events news (1), special-sparts (1), specialsservice (1), specialsservice department (1), system (12), technivision (1), tool (4), type (10), upcoming events news (1), vehicle (10), wagner (1), wagner jaguar (1)}

Table 4. Cluster DRM for the entity Jaguar

Cluster #	3	1	2	4
DRM	0,070	0,011	0,000	0,000

Step 2.1: Identify domain relevant clusters

By calculating the similarity with the clusters of the neighbouring entities of **Jaguar**, which are **Felidae** (super-class) we can identify the relevant cluster for this domain. In this case, **Cluster#3** had the highest similarity (see Table 4) with a DRM score of 0,070. This cluster is therefore selected as the candidate cluster for the construction of the feature vector to be done in the next step.

Step 2.2: Construct feature vector

The last step for the **Jaguar** entity is to create the final entity feature vector, which in this example will be the same as the CLFVs for **Cluster#3** as seen in Table 5. At this stage, we could do a more thorough analysis of the cluster documents to improve the quality of the feature vector even further.

Table 5. The final fv created for the **Jaguar** entity

Jaguar={cat (26), culture (11), habitat (13), jaguar (164), panthera (11), population (11), prey (19), range (20), species (27), state (11)}

4 Experiments

We have conducted a set of experiments (described in Section 4.2) to validate the feature vector construction algorithm discussed in Section 3. The goal of the experiments is to measure the sensitivity both w.r.t. some of the components of the approach and some ontologies of different granularity (presented in Section 4.1). We are using Normalized Google Distance (NGD) (described in Section 4.3) and two additional measures to get a representative value of the feature vector quality. In Section 5, we will present and discuss the results of the experiments.

4.1 Ontologies

FVs' construction is semantics based and heavily relies on ontologies. Consequently, we would like to measure the effect of ontologies of different granularity. We have chosen three ontologies that have been used in our earlier experiments [18]. All the ontologies are formalized in OWL DL. Next, short descriptions of the ontologies⁶ are provided:

Animals ontology: this little ontology classifies some species, does not contain any individuals, and has only hierarchical properties. The original ontology is adapted to be more correct w.r.t. biological classification. The ontology was selected to see the effect of applying the approach on a typical *taxonomy*.

Travel ontology: A bit more advanced compared to the Animals ontology by having individuals and some object properties. This ontology is classified in this work as a *lightweight* ontology.

Wine ontology: Even more advance than the Travel ontology with more individuals than classes and many relations. This ontology was originally constructed to test reasoning capabilities. Maybe as a result, the ontology contains some entity labels that are not found elsewhere (e.g. the entity *McGuinnesso* is according to the ontology a winery; however a search with Google provides no results). Consequently, there will be several entities that will not be populated with this ontology. This ontology can indicate the robustness of this approach and is classified in this work as *advanced*.

We have selected not to include any large or *heavyweight* ontologies in this experiment since we believe that larger ontologies will not provide any significant new insight except of processing time, which is not the focus of this evaluation.

The key characteristics of the ontologies are displayed in Table 6. The evaluation has restrictions as follows:

- All OWL object properties are treated as other relations.
- Disjoint classes as a feature are ignored since we do not consider siblings in this evaluation.
- The following equality features are ignored: `equivalentClass`, `sameAs`, and `differentFrom`.
- No reasoner is used. A reasoner can be used to extract more relationships between the entities than are available without using a reasoner. These additional relationships can be utilized to improve the FV quality.
- The maximum length of the FVs has been set to 30. In earlier experiments [18], the average FV length was 24 ± 3 .
- For query expansion, there have been set a limitation of maximum 3 entities from each of the possible neighbour relation types (parents, children, and others), that implies query expansion by maximum 9 entities in total.

Table 6. Ontology key characteristics

Ontology	Classes	Individuals	Properties
Animals	51	0	0
Travel	34	14	6
Wine	82	155	10

4.2 Experimental configurations

In this section, we will describe the experiments and the motivation behind them. The conducted experiments are summarised in Table 7. Next, we briefly describe each of the experiments.

Baseline (BI#1, 2): A baseline was created in order to compare the results. For the domain identification component (Step 2.1), we selected to use *parent* entities for comparison since it

⁶ The ontologies used can be found here: <http://folk.ntnu.no/steint/ontologies/>

must compare with at least one neighbouring entity. The baseline was conducted twice: at the beginning and at the end of the experiments. This was done in order to isolate influence of time span (see Section 5). The experiments were conducted in a period of one week.

Query expansion - neighbours (Ex#2-8): We test what kind of neighbouring entities (parent, child, other) are optimal to include.

Query expansion - keywords (Ex#12, 13): By populating an ontology with global keywords it is expected that all the FVs will have higher similarity and be less unique compared to omitting the global keywords. However, is this the case?

Number of search results (Ex#14, 15): 30 search results have been set for the baseline. Is this an optimal number and what implication has it on the FV quality? We test if 100 or even 200 are more optimal. We expect that more search results will have a positive effect on the FV quality.

Content (Ex#9): It is expected that using full text documents will provide better feature vector quality than using snippets.

Clustering - input (Ex#10, 11): The clustering algorithms used are optimized for processing snippets. As a result, it is assumed that using document feature vectors will be a better candidate than using raw full text documents. However, for snippets it might be better to use the raw text than creating document feature vectors since snippets do in general provide little information and if only some of the key-phrases are extracted then even less information will be available to the clustering algorithm.

Domain identification (Ex#16-21): It is expected that comparing with neighbouring entities by relation type filtering will have a major effect on the feature vector quality. Utilizing parents are assumed in general to have the most positive effect.

Best practice (Ex#22): As the experiment proceeded we started to get some indications of what components and parameters that had a positive effect on the feature vector quality or not. Consequently, we would also like to test if a combination of these findings would yield the same positive effect or not. Therefore, we have combined some of these findings to assess the effect.

Table 7. Summary of the experiments conducted

	BI#1	BI#2	EX#2	EX#3	EX#4	EX#5	EX#6	EX#7	EX#8	EX#9	EX#10	EX#11	EX#12	EX#13	EX#14	EX#15	EX#16	EX#17	EX#18	EX#19	EX#20	EX#21	EX#22	
0. Ontology																								
Animals	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Travel	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Wine	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
1. Query expansion																								
<i>neighbors</i>																								
parents			X			X	X		X					X										X
children				X		X		X	X															X
others					X		X	X	X															X
keywords													X ¹	X ¹										
2. Search results																								
<i>content</i>																								
snippet	X	X	X	X	X	X	X	X	X		X		X	X	X	X	X	X	X	X	X	X	X	X
full text										X		X												
nbr of results	30	30	30	30	30	30	30	30	30	30	30	30	30	30	100	200	30	30	30	30	30	30	30	100
3. Clustering																								
<i>input</i>																								
document fv											X	X												X
text	X	X	X	X	X	X	X	X	X	X			X	X	X	X	X	X	X	X	X	X	X	
4. Domain identification																								
<i>neighbors</i>																								
parents	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X				X	X		X	X
children																	X		X		X	X	X	
others																		X		X	X	X		

¹ Animals ontology: 'animals'; Travel ontology: 'travel'; Wine ontology: 'wine'

4.3 Evaluation measures

In this section, we will define the similarity measures used. First, we define the Average Fv Similarity (AFvS) as follows.

Definition 3. *Average Fv Similarity (AFvS)* gives an indication of the uniqueness of the FVs.

$$AFvS(o) = \frac{2}{n^2 - n} \sum_{i=1}^n \sum_{j=i+1}^n sim(fv_i, fv_j) \quad (3)$$

where n is the number of fvs in the ontology o and $sim(fv_i, fv_j)$ is the traditional cosine similarity measure between the two vectors. A score of zero would indicate that all FVs are unique. However, this is hardly possible since the approach is based on similarity between the entities to be able to populate the ontology. In general we would like this score to be as low as possible, in order to discriminate the entity FVs. However, this depends a lot on ontology.

Next similarity score is the Average Fv Neighbourhood Similarity (AFvNS) defined as follows.

Definition 4. *Average Fv Neighbourhood Similarity (AFvNS)* indicates the degree of overlap with neighbouring entities.

$$AFvNS(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m sim(fv_i, fv_j) \quad (4)$$

where n is the number of fvs in the ontology o and m is the number of neighbouring entities with fvs of entity i with fv_i . In this experiment, we have selected to use all the neighbours of an entity and do not differentiate the neighbours by weighting. As for AFvS this score will be highly depended on the ontology quality. Nevertheless, the ideal score would depend on the intended usage of the populated ontology (e.g. when used in search, for a comprehensive search we would like this value to be higher than for a fact-finding kind of search).

We have chosen to use the Normalized Google Distance (NGD) [6] as a measure to evaluate the quality of each feature vector. NGD can be used to compute the semantic distance between two terms. The NGD equation [6] is provided below for the clarity:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (5)$$

where $f(x)$ denotes the number of pages containing x and $f(y)$ for y , and $f(x, y)$ denotes the number of pages containing both x and y . N denotes the "total number" of pages assumed index by Google, which in this experiment was set to 20 billion (at this magnitude the precise amount of pages is not significant). The range of NGD is between 0 and ∞ , where 0 denotes best match. However, in practice most values are in the range from 0 to 1. Consequently, for the special case where $NGD(x, y) > 1$ we set $NGD(x, y) = 1$. The motivation behind this is that the distance is too large to be of any interest anyway. Note, for this assumption to be valid the constant N must be set to a representative value. NGD is symmetric by definition, however searches with Google are not (e.g. a search for "x y" often yield different results than "y x"). We tackle this issue by ordering the search term (for instance, always putting the parent entity before a child entity).

NGD will be used in the next similarity scores as follows.

Definition 5. *Average Fv NGD (AFvNGD)* indicates the semantic distance between the entities and their FVs.

$$AFvNGD(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m NGD(fv_i, kp_j) \quad (6)$$

where n is the number of fvs in the ontology o and m is the length of the fv_i and fv_i is the name of the fv_i , the entity name, and kp_j are the key-phrases of fv_i . Note if an entity got a parent then the name of the parent is also included to provide a more specific similarity distance

(adapted from Bouquet et al. [3] that in our case is limited to the closest parent). $FvNGD(fv)$ will have a score in the same range as NGD.

Once we have found the AFvS, AFvNS and the AFvNGD measures for an ontology the total score can be calculated. The total score is an aggregated score of the three measures. The total feature vector quality score is defined as follows.

Definition 6. *Fv Quality Score (FvQS)* provides the overall quality of the FVs.

$$FvQS(o) = \alpha(1 - AFvS) + \beta AFvNS + \gamma(1 - AFvNGD) \quad (7)$$

where $\alpha + \beta + \gamma = 1$ are weight factors (defaults are 1/3). The total FV quality score for an ontology will be in the range (0-1), where 1 indicates the best score.

5 Results and analysis

In this section, the results of the experiments are presented and analysed. Note, because of constant change of the Web corpora and update of search engines the results of this evaluation may vary in time. Therefore, the evaluation was conducted in a week to minimize the issue of results changing due to changes provided by the search engine providers.

Table 8. Experimental results

	<i>Animals ontology</i>				<i>Travel ontology</i>				<i>Wine ontology</i>			
	AFvS	AFvNS	AFvNGDS	FvQS	AFvS	AFvNS	AFvNGDS	FvQS	AFvS	AFvNS	AFvNGDS	FvQS
BI#1	0,019	0,154	0,266	0,623	0,019	0,186	0,253	0,638	0,040	0,286	0,163	0,694
BI#2	0,020	0,168	0,255	0,631	0,023	0,147	0,253	0,624	0,041	0,286	0,180	0,688
Ex#2	0,048	0,304	0,194	0,687	0,042	0,326	0,227	0,686	0,079	0,412	0,149	0,728
Ex#3	0,021	0,288	0,277	0,663	0,021	0,313	0,254	0,679	0,046	0,322	0,155	0,707
Ex#4	0,021	0,178	0,265	0,631	0,020	0,139	0,241	0,626	0,041	0,304	0,152	0,704
Ex#5	0,040	0,404	0,214	0,717	0,035	0,243	0,231	0,659	0,075	0,403	0,150	0,726
Ex#6	0,048	0,288	0,200	0,680	0,041	0,334	0,231	0,687	0,079	0,409	0,149	0,727
Ex#7	0,020	0,278	0,276	0,661	0,021	0,259	0,258	0,660	0,043	0,316	0,158	0,705
Ex#8	0,039	0,406	0,215	0,717	0,034	0,272	0,233	0,668	0,073	0,412	0,149	0,730
Ex#9	0,015	0,211	0,261	0,645	0,049	0,239	0,246	0,648	0,102	0,458	0,192	0,722
Ex#10	0,019	0,130	0,270	0,613	0,019	0,092	0,241	0,611	0,042	0,277	0,177	0,686
Ex#11	0,014	0,224	0,249	0,654	0,049	0,225	0,229	0,649	0,099	0,446	0,196	0,717
Ex#12	0,182	0,280	0,262	0,612	0,161	0,253	0,243	0,616	0,286	0,452	0,182	0,661
Ex#13	0,133	0,358	0,197	0,676	0,098	0,261	0,220	0,648	0,218	0,467	0,170	0,693
Ex#14	0,017	0,210	0,259	0,644	0,022	0,201	0,241	0,646	0,045	0,386	0,184	0,719
Ex#15	0,015	0,221	0,268	0,646	0,026	0,233	0,249	0,652	0,054	0,397	0,185	0,720
Ex#16	0,022	0,070	0,149	0,633	0,029	0,177	0,249	0,633	0,079	0,345	0,208	0,686
Ex#17	-	-	-	-	0,014	0,195	0,192	0,663	0,048	0,293	0,268	0,659
Ex#18	0,018	0,181	0,260	0,635	0,026	0,136	0,236	0,625	0,045	0,308	0,156	0,703
Ex#19	0,019	0,180	0,259	0,634	0,025	0,132	0,226	0,627	0,043	0,307	0,152	0,704
Ex#20	0,010	0,030	0,150	0,623	0,023	0,151	0,258	0,623	0,059	0,337	0,241	0,679
Ex#21	0,018	0,172	0,231	0,641	0,023	0,137	0,234	0,627	0,042	0,311	0,180	0,696
Ex#22	0,044	0,487	0,198	0,749	0,043	0,343	0,237	0,687	0,101	0,553	0,174	0,759

5.1 Results and analysis

Table 9 summarises the test results where the evaluation measures described in Section 4.3 were used. In total 23 experiments were conducted. The first experiment conducted was BI#1 while the last was BI#2 that are used as the baseline for the other experiments. In the next section, we will analyse the results.

Table 9. Experimental analysis

	Bl#1	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10	Ex#11
Animals ontology	0,0%	9,1%	5,3%	0,0%	13,8%	7,9%	4,8%	13,9%	2,3%	-2,8%	3,7%
Travel ontology	0,0%	9,7%	8,7%	0,4%	5,5%	10,0%	5,7%	7,0%	3,8%	-2,0%	4,0%
Wine ontology	0,0%	5,7%	2,7%	2,2%	5,5%	5,6%	2,4%	6,0%	4,8%	-0,3%	4,1%
Average	0,0%	8,2%	5,6%	0,9%	8,3%	7,8%	4,3%	9,0%	3,6%	-1,7%	4,0%
Standard deviation	0,0%	2,2%	3,0%	1,2%	4,8%	2,2%	1,7%	4,3%	1,2%	1,3%	0,2%

	Ex#12	Ex#13	Ex#14	Ex#15	Ex#16	Ex#17	Ex#18	Ex#19	Ex#20	Ex#21	Ex#22
Animals ontology	-3,0%	7,3%	2,2%	2,4%	0,4%		0,6%	0,6%	-1,2%	1,7%	18,9%
Travel ontology	-1,1%	3,8%	3,5%	4,5%	1,5%	6,2%	0,2%	0,5%	0,0%	0,5%	10,0%
Wine ontology	-3,9%	0,7%	4,4%	4,5%	-0,3%	-4,2%	2,1%	2,2%	-1,3%	1,1%	10,2%
Average	-2,7%	3,9%	3,4%	3,8%	0,5%	1,0%	1,0%	1,1%	-0,9%	1,1%	13,1%
Standard deviation	1,4%	3,3%	1,1%	1,2%	0,9%	7,3%	1,0%	1,0%	0,7%	0,6%	5,1%

An overview of the experiments and their percentage difference relative to the baseline is shown in Table 9. Since we used Bl#1 as the baseline the values for this experiment is set to 0. Further, since we are using the Web and depends on search results from a commercial search engine, where we have little control of potential changes that might affect the search results, we conducted the same baseline test as the final test of these experiments. This new baseline test is denoted as Bl#2. Consequently, Bl#2 serves as deviation value and therefore subtracted from the results shown in Table 9. Next, we will provide some comments about the findings of the experiments:

Query expansion - neighbours (Ex#2-8): Ex#8 provided in average the best results, and also the best results for the Animals and the Wine ontologies. However, for the Travel ontology Ex#8 provided the fourth best results while Ex#6 gave the best results for this ontology. It was assumed that Ex#2 in average would provide the best results however it turned out that it provided the third best results. If we look at both the standard deviation and mean results then Ex#2 yields the best results. This could indicate that independent of the quality of the ontology Ex#2 would be the best choice.

Query expansion - keywords (Ex#12, 13): The results from Ex12# indicate that adding global keywords is not beneficial w.r.t. the overall FV quality score. The AFvS score is high for both Ex#12 and Ex#13. Ex#13 indicates an increase but compared to Ex#2 it is a decrease. However, as discussed in Section 3 Step 1.1, homogeneous FVs can be a feature that is beneficial depending on the intended usage.

Number of search results (Ex#14, 15): In Ex#14 and Ex#15 we tested if the number of search results retrieved and processed would affect the FV quality, which provide to be the case with 3.4% and 3.8% respectively. More clusters are more expensive to compute. In Ex#2, the Animals, Travel and Wine ontologies took 3, 3, and 16 minutes to process respectively while Ex#14 took in average 3 times as long to process and Ex#15 took 7 times as long. In this experiment we have not tried to find the optimal number of results to process, but just by looking at the increase of FV quality from 30 to 100 results versus 200 results indicate that 100 is the best candidate in this test w.r.t. both the FV quality and processing time.

Content (Ex#9): The results of Ex#9 show a slight improvement with an average of 3.0% compared to the baseline. It is uncertain if this result is optimal since we have experienced some difficulties using full text documents. Many sites do not allow direct download of Web pages for other purposes than browsing. Consequently, some of the documents became unavailable which would influence the quality of the FVs. Nevertheless, Ex#9 showed an improvement compared to the baseline.

Clustering - input (Ex#10, 11): In Ex#11 we tested if it is more beneficial to use document FVs, key-phrases extracted from the full text documents, as input to the clustering algorithms or snippets. Ex#11 showed some improvement of using document FVs compared to Ex#9 with only 0.4%, probably because the document FVs are more focused by extracting only those parts of the documents considered most relevant to the search. However, when creating document FVs for the snippets, Ex#10 showed a decrease in performance by 1.7% indicating that the snippets are best used as is.

Domain identification (Ex#16-21): Not surprisingly we got more or less the same results as for the query expansion experiments (Ex#2-Ex#8) where using *parents*, *children*, and *other* neighbouring entities provided the best results (Ex#21). Ex#19 got the same results as Ex#21 but with higher standard deviation indicating that Ex#21 provides better results independent of the ontology quality. Ex#18 and Ex#19 provides more or less same results. For Ex#16, Ex#17, and Ex#20 the algorithm failed to populate most of the entities (see Table 10). In fact, for Ex#17 no entities were populated for the Animals ontology since the ontology only got super- and sub-class relationships and hence no *other* relations. Consequently, the results from Ex#16, Ex#17, and Ex#20 can be disregarded.

Best practice (Ex#22): These experiments were conducted to test the combination of some of the best results from the other experiments. Both Ex#22 performed considerably better than the other experiments with an increase of 13.1% and 10.6% respectively.

5.2 Key findings

Based on the findings in the conducted experiments we conclude the following:

(1) *Query expansion*: Query expansion increases the quality of the search results and hence the quality of the FV quality. Including the parents, children, and other related entities provide the best results.

(2) *Search results* and (3) *Clustering*: Using full text documents in combination with extraction of the most relevant key-phrases seems to provide the best positive effect on the FV quality. However, this increases the processing time considerably compared to using just snippets (assumes this is mainly due to download of each page).

(4) *Domain identification*: Including the parents, children, and other related entities seem to provide the best results when identifying the most prominent cluster candidates.

However, these are general conclusions independent of ontology quality. The most important component with respect to the FV quality is the query expansion component (Step 1.1). The parent entities are the most important neighbouring entities both for query expansion (Step 1.1) and when identifying the most prominent candidate cluster (Step 2.1). Further, utilizing the neighbouring entities when expanding the query yields better FV quality than using scope keywords. A high number of search results minimises the difference between the search engines and probably the change in ranking they provide over time.

6 Conclusions and future work

In this study, we have described and evaluated an unsupervised approach to feature vector construction. These feature vectors typically contain terms that are associated with the concepts reflected by the actual text corpora, i.e. the Web. We have focused on the aspects of the components w.r.t. both the FV quality and the ontologies used. Ontologies with different granularity where used, we have shown how this affect the quality of the feature vectors. In total 23 experiments were conducted. Based on the findings a set of recommendations for the construction of ontology based feature vectors are proposed.

We have also done some minor experiments with the NGD measure to assess the semantic distance between the entities of the ontologies used in this experiment. Preliminary results indicate that there is a connection between the findings and characteristics of each ontology used in this experiment and the NGD ontology score. This needs to be explored further. Therefore, one of the future tasks is to conduct a similar experiment with a broader set of ontologies. We need to categorize the ontologies according to different key characteristics to find trends relevant to the categories.

References

- [1] Agirre, E., Ansa, O., Hovy, E.H., Martínez, D.: Enriching very large ontologies using the WWW. ECAI Workshop on Ontology Learning, Vol. 31. CEUR-WS.org (2000)
- [2] Bergamaschi, S., Bouquet, P., Giazomuzzi, D., Guerra, F., Po, L., Vincini, M.: An Incremental Method for the Lexical Annotation of Domain Ontologies. *Int. J. on Semantic Web and Information Systems* 3(3) (2007) 57-80
- [3] Bouquet, P., Serafini, L., Zanobini, S.: Semantic Coordination: A New Approach and an Application. *The SemanticWeb - ISWC 2003* (2003) 130-145
- [4] Bry, F., Koch, C., Furche, T., Schaffert, S., Badea, L., Berger, S.: Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages. *Int. J. on Semantic Web and Information Systems* 1(2) (2005) 1-21
- [5] Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. *IEEE TKDE* 19(2) (2007) 261-272
- [6] Cilibrasi, R. and Vitanyi, P. The Google Similarity Distance. *IEEE Transactions on Knowledge and Data Engineering*, 19 (3). 370-383.
- [7] Formica, A., Missikoff, M., Pourabbas, E., Taglino, F.: Weighted Ontology for Semantic Search. Proceedings of the OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008. Part II on On the Move to Meaningful Internet Systems. Springer-Verlag, Monterrey, Mexico (2008) 1289-1303
- [8] Justeson, J.S., Katz, S.M.: Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, Vol. 1. Cambridge University Press (1995) 9-27
- [9] Lopez, V., Sabou, M., and Motta, E.: PowerMap: Mapping the Real Semantic Web on the Fly. *ISWC 2006*, LNCS 4273, 414 – 427, 2006.
- [10] Mitchell, T.M.: *Machine Learning*. McGraw-Hill, New York (1997)
- [11] Ogden, C.K., Richards, I.A.: *The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism*. Kegan Paul, Trench, Trubner & Co, London (1930)
- [12] Osinski, S., Weiss, D.: A Concept-Driven Algorithm for Clustering Search Results. *IEEE Intelligent Systems*, Vol. 20 (2005) 48-54
- [13] Panagis, Y., Sakkopoulos, E., Garofalakis, J. and Tsakalidis, A. Optimisation mechanism for web search results using topic knowledge. *Int. J. Knowledge and Learning* 2(1/2) (2006) 140–153
- [14] Solskinnsbakk, G., Gulla, J.: *Ontological Profiles in Enterprise Search*. Knowledge Engineering: Practice and Patterns (2008) 302-317
- [15] Strasunskas, D., Tomassen, S.L.: The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation. *Int. J. Knowledge and Learning* 4 (2008) 398-414
- [16] Su, X., Gulla, J.A.: An information retrieval approach to ontology mapping. *Data & Knowledge Engineering* 58 (2006) 47-69
- [17] Suomela, S., Kekalainen, J.: Ontology as a search-tool: A study of real user's query formulation with and without conceptual support. In: *Proceedings of ECIR'2005*. LNCS 3408, Springer-Verlag (2005) 315-329
- [18] Tomassen, S.L. and Strasunskas, D.: *Construction of Ontology based Semantic-Linguistic Feature Vectors for Searching: the Process and Effect*. *WI-IAT '09*, IEEE Computer Society, Milano, Italy, 2009
- [19] Yang, H-C. A method for automatic construction of learning contents in semantic web by a text mining approach. *Int. J. Knowledge and Learning* 2(1/2), (2006) 89–105
- [20] Zhou, X., Hu, X., Zhang, X.: Topic Signature Language Models for Ad hoc Retrieval. *Knowledge and Data Engineering*, IEEE Transactions on 19 (2007) 1276-1287

P3: Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation

Publication details

Tomassen, S.L. & Strasunskas, D. (2009) Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation. In: Kotsis, G., Taniar, D., Pardede, E. & Khalil, I. (eds.) *Proceedings of the 11th Int. Conf. on Information Integration and Web-based Applications & Services*, ACM, New York, pp. 86-93.

Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation

Stein L. Tomassen

IDI, NTNU

Sem Saelandsvei 7-9,
NO-7491 Trondheim, Norway
+ 47 735 94218

stein.l.tomassen@idi.ntnu.no

Darijus Straszunas

IOT, NTNU

Alfred Getz veg 3,
NO-7491 Trondheim, Norway
+ 47 735 93659

darijuss@gmail.com

ABSTRACT

Search is among the most frequent activities on the Web. However, the search activity still requires extra efforts in order to get satisfactory results. One of the reasons is heterogeneous information resources and exponential growth of information. In this paper we try to tackle these issues. We elaborate on an approach to construction of semantic-linguistic feature vectors (FV) that are used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. The value of this approach is twofold. First, it captures relevant semantics from an ontology, and, second, it accounts for statistically significant collocations of other terms and phrases in relation to the ontology entities. In this paper, we elaborate on the extended FV construction process and evaluate the FV quality with respect to a set of heterogeneous ontologies. The evaluation shows that ranking of entities is significant neither for FV quality nor FV construction process. However, the results demonstrate that the construction process is most sensitive to taxonomy type of ontologies while usage of advanced and rich ontologies produces better quality FVs.

Categories and Subject Descriptors

H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing - *abstracting methods, linguistic processing*.

H.3.4 [Information Storage and Retrieval]: Systems and Software - *performance evaluation (efficiency and effectiveness)*.

Keywords

Ontology, Feature Vector Construction, Evaluation.

1. INTRODUCTION

Nowadays, the Web is one of the dominant information sources for learning and acquiring new knowledge. However, finding the relevant information is still a huge challenge. The emerging Semantic Web (SW) will eventually solve some of the problems. However, search needs to be improved now.

Improvement can be achieved by combining strengths of the current Web search with the emerging semantic

techniques. This endeavour is referred to as semantic search that is differentiated from the querying of the SW. This can be achieved by a combination of various techniques: semantic annotations (e.g., [3, 17]); clustering of retrieved documents according to topics (e.g., [11]); powerful querying languages (e.g., [5]). In summary, many efforts are devoted to improve information retrieval (IR) using ontologies (e.g., [7, 14]).

The objective of this paper is to present and evaluate the proposed approach to semantic search that builds on a concept of *feature vectors* (FV). The approach is based on pragmatic use of ontologies by relating the concepts (domain semantics) with the actual terminology used in a text corpus, i.e. the Web. Therefore, we propose to associate every entity (classes and individuals) of the ontologies with a FV to tailor them to the domain terminology in a text corpus. First, these FVs are created off-line and later used on-line to filter, and hence disambiguate search, and re-rank the search results from the underlying search system [12].

In this paper, we focus on the FV construction process since the actual search performance depends a lot on the quality of FVs. In [12], we investigated FVs use in search disambiguation that was evaluated with real users. While in [15], the FVC algorithm used in [12] was elaborated. In [12] we found significant dependence between overall performance and ontology quality, however we were not able to conclude to what degree FV quality depends on ontology and how much it is influenced by FV construction process and techniques used there. Therefore, in [16] we focused on aspects of the components of the algorithm presented in [15] that affect the FV quality. The contribution of this paper is an extended version of the algorithm where ranking of entities are used.

This paper is organized as follows. In section 2, related work is discussed. In section 3, the algorithm of how the FVs are constructed and a small example of the process are presented. In Section 4, we present the conducted experiments and explain the evaluation. Then in section 5, the results will be analyzed. Finally, in section 6, we conclude this paper.

2. RELATED WORK

In this section, we provide an overview of related work on enhancement of search by semantics. The literature review is limited to approaches that build on a notion similar to our feature vectors.

The approach by Su and Gulla [13] constructs FVs for all the concepts of an ontology, which are used for ontology mapping and not for semantic search. Nevertheless, the process of constructing the FVs has similarities to our approach. The main differences are: the documents are assigned semi-automatically; an initial highly relevant document collection is necessary; and the FVs are constructed by taking the average of all assigned document vectors. This in contrast to our approach, which is using the Web as text corpus and use knowledge represented in the ontologies to find the most relevant documents and associated terms.

Agirre et al. [1] present Topic Signatures (TS) that are used to enrich WordNet. TS are vectors with terms being related to a topic, equal to our FVs. The vectors are created in a similar fashion to ours but depend on specifying highly relevant queries to avoid noisy TS in contrast to our approach where this is not necessary. Consequently, the biggest challenge with the approach is specifying queries that return neither too few (or none) nor too many results. The queries are created using the hierarchical information provided by WordNet. The approach was evaluated by word disambiguation tasks showing good results.

Finally, Gabrilovich and Markovitch [10] utilize the vast amount of organized human knowledge that is available in knowledge repositories like Wikipedia and Open Directory Project (ODP). Each node in ODP is treated as a concept. A textual object is created for each node consisting of concatenated Web documents (listed for each node by ODP) and their textual descriptions. The concepts are represented as attribute vectors. A document is divided into non-overlapping segments called contexts where each context is related to one or several concepts. An ambiguous concept will be part of several subject domains (contexts) that is partly resolved by categorization of the contexts. In case of hierarchies, a parent node will typically consist of both the child concepts and a textual description, similar to Su and Gulla [13].

3. FEATURE VECTOR CONSTRUCTION

Every ontology entity has a feature vector with a set of associated terms extracted from a text corpus. An entity can be either a class or an individual. In this section, we will describe the process of how these FVs are constructed, but first we provide a definition of a FV. At the end of this section, we exemplify the construction process.

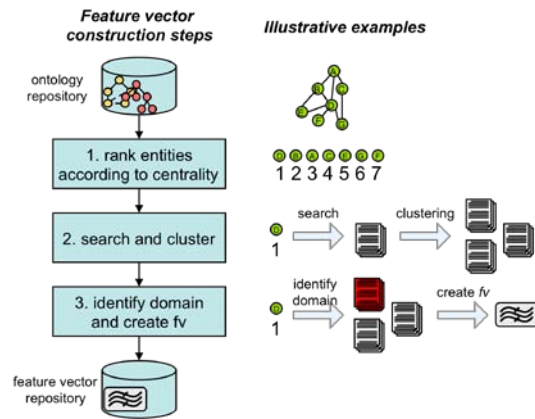


Figure 1. An overview of the FV construction process.

3.1 Introduction to Feature Vectors

A feature vector "connects" a concept (entity) to a document collection (i.e. the FV is tailored to the specific terminology used in a particular document collection). FVs are built considering both semantics encoded in an ontology and the dominant lexical terminology surrounding the concepts (entities) in a text corpus. The underlying idea is that a FV reflects both the semantic and linguistic neighbourhoods of a particular entity. The semantic neighbourhood is computed based on related entities and direct properties specified in an ontology, while the linguistic neighbourhood is based on the co-location of terms in a domain specific corpus. Therefore, a FV constitutes a rich representation of an entity that is related to the actual terminology used in a text corpus. For a more formal definition of a FV, the keen reader is referred to [16].

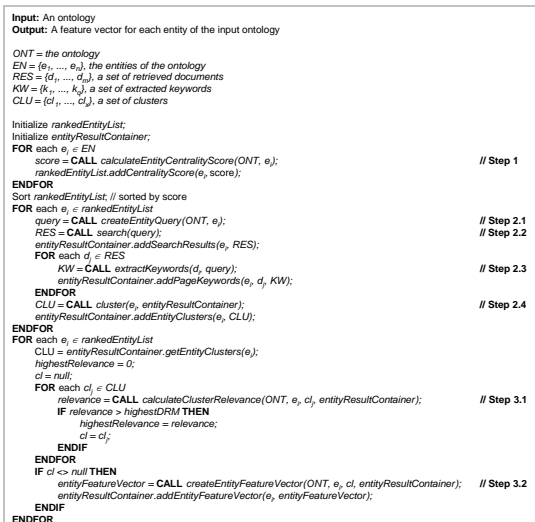


Figure 2. The Feature Vector Construction algorithm.

3.2 Construction of Feature Vectors

The feature vector construction (FVC) process is composed from three phases (main steps) (see Figure 1). The FVC process presented in [16] contained two phases (the two last phases of the algorithm presented here). The first phase includes ranking of the ontology entities according to their importance w.r.t. the ontology, this helps to optimize phase 3. The main aim of phase 2 is to extract and group sets of candidate terms being relevant to each entity. However, the candidate terms are not necessarily relevant to the domain defined by the ontology. Consequently, the aim of the last phase is to identify those candidate terms being relevant to the entities defined by the ontology. Finally, a FV for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding FVs that consist of terms associated to both the entities and the domain terminology. Below, we elaborate each of the steps as follows.

Step 1: Rank entities

Since we endeavour to create FVs for every entity (i.e. both a class and an individual) in the ontology, the algorithm starts with traversing the ontology and ranks each entity according to relevancy. The result of this process is a ranked list of entities according to considered importance (centrality) w.r.t. the ontology. This list of ranked entities is later used to identify those documents being relevant to the domain defined by the ontology (Step 3). A ranked list versus a random list of entities is believed to improve the quality of identifying the most relevant candidate terms done in Step 3. The idea is that more information is available for the most central entities, the better opportunities to discriminate relevant candidate terms. Those entities that already have been assigned relevant terms are later used to identify the most relevant candidate terms for other entities (more details in Step 3).

We have adapted the AKTiveRank algorithm by Alani et al. [2] to rank the entities. The original intention of AKTiveRank is to rank several ontologies for comparison. However, some of the measures are suitable to measure the centrality of entities w.r.t. the ontology. Consequently, we have focused on those elements of the algorithm, which are the class betweenness measure being part of the Betweenness Measure (BEM) and the class density measure being part of the Density Measure (DEM). BEM gives an indication of the centrality of an entity in the sense of where it is graphically located within an ontology. The centrality is found by calculating the number of shortest paths that pass through each entity of the ontology. Our definition of Entity Betweenness Measure (EBM) is equal to the $bem(c)$ definition of BEM [2] and is as follows.

Definition 1: Entity Betweenness Measure (EBM)

Let $e_i, e_j \in \{E[O]\}$, e_i and e_j are any two entities in the ontology O . $E[o]$ is the set of entities in ontology o .

$$EBM(e) = \sum_{e_i \neq e_j, \forall e \in E[o]} \frac{\sigma_{e_i e_j}(e)}{\sigma_{e_i e_j}} \quad (1)$$

where $EBM(e)$ is the Entity Betweenness Measure for entity e . $\sigma_{e_i e_j}$ is the shortest path from e_i to e_j , and $\sigma_{e_i e_j}(e)$ is the number of shortest paths from e_i to e_j that passes through e .

For the Entity Density Measure (EDM) we have adopted the class density measure by Alani et al. [2]:

Definition 2: Entity Density Measure (EDM)

Let $S = \{S_1, S_2, S_3, S_4, S_5\} = \{\text{sub-classes}[e], \text{super-classes}[e], \text{individuals}[e], \text{siblings}[e], \text{relations}[e]\}$

$$EDM(e) = \sum_{j=1}^5 w_j |S_j| \quad (2)$$

where $EDM(e)$ is the Entity Density Measure for entity e and w_j is a weight factor with default value of 1.

Then the total Entity Centrality Score (ECS) for an entity is calculated using both EBM and EDM:

Definition 3: Entity Centrality Score (ECS)

$$ECS(e) = \alpha EBM(e) + \beta EDM(e) \quad (3)$$

where $ECS(e)$ is the Entity Centrality Score for entity e and $\alpha + \beta = 1$ are the centrality and density weight factors respectively. Both α and β is set to a default value of 0.5.

A pre-ranking of the entities is achieved when ECS is calculated for all the entities of the ontology. However, the current order of this list does not ensure that e.g. the second entity is directly connected to the first entity. Hence, we need a ranked list of entities that is based on both centrality and where each next entity in the list directly connects with any of the prior entities of the list. This sought list of ranked entities is assured by using the Spreading Activation algorithm [9]. First, the entity with the highest ECS is selected. If there are several entities with equal score then the sum of the neighbour entities' ECS is calculated. The entity with the highest score is selected. The selected entity will act as the initial node of the Spreading Activation algorithm. The Spreading Activation algorithm ends when there are no more entities left in the pre-ranked list. Entities with no direct relation(s) to other entities will be omitted since those entities are identified as loners (e.g. an entity only being a subclass of `owl:Thing`). This FVC algorithm is not able to associate FVs for loners since neighbouring entities are vital in the process of identifying highly relevant terms (more of this in Step 3). The result of this step is a ranked list of entities that is based on both centrality and density of the ontology.

Step 2: Search and cluster

This step constitutes three sub-steps where the aim is to extract and group sets of candidate terms being relevant to each entity.

Step 2.1: Compose entity query

A query for each entity (Step 1) is created and used in Step 2.2 to retrieve candidate documents for each entity. The query is based on the entity name and optionally expanded with selected neighbouring entities and/or keywords. Neighbouring entities can be *parent*, *child*, and/or *other*. A *parent* entity of a class is defined to be its super-class, while a *parent* of an individual is the class the individual being an instance of. A *child* entity of a class is defined to be its sub-class or individual, the latter if it does not have a sub-class. Finally, *other* neighbouring entities are any other object property defined in OWL.

Step 2.2: Entity based search

The queries created in Step 2.1 are used to retrieve a set of candidate documents for each entity. Any search engine can be used in this step. In the experiments described in Section 4.2 Yahoo!™ is used. The retrieval session is keyword-based.

Step 2.3: Contextual key-phrase extraction

For each document a set of key-phrases and keywords is extracted, hereinafter referred to as key-phrases. First, a part of speech (POS) tagger is used to tag the retrieved documents (snippet or full text). Then a set of 39 tagging rules [15] is applied. Based on these rules a set of candidate key-phrases are extracted. However, only those key-phrases that are within the contextual windows (i.e. frames surrounding the entities) are extracted. Each key-phrase is stemmed to remove potential duplicates. Finally, those candidate key-phrases above a frequency threshold are kept and stored in a Document Feature Vector (DFV) for that document.

Step 2.4: Cluster search results

In order to identify (discriminate) different domains (by documents) found for each entity, clustering techniques are used. At this stage of the process, the ontology entities are treated as ordinary terms (words) and consequently can be used in many different domains. Clustering allows finding different domains by grouping similar documents (the most relevant domain w.r.t. the ontology is identified in Step 2.1). Currently the Carrot API [6] is used. The result of this step is a set of clusters for each entity. In addition, for each cluster a Cluster Feature Vector (CLFV) is created. A CLFV is a combination of all the DFVs associated with the documents (created in Step 2.3) of the cluster.

Step 3: Identify and construct

This step is constituted of two sub-steps and aims to identify the most relevant clusters w.r.t. the ontology and create the final FVs.

Step 3.1: Identify domain relevant clusters

A problem at this stage is to identify the correct domain, that is, the most relevant clusters found in Step 2.4 w.r.t. the ontology. Therefore, we compute the similarity between the CLFVs of an entity with the CLFVs of its neighbouring entities.

Commonality (i.e. high similarity) here identifies the document sets (clusters) being relevant to the domain of our interest. The hypothesis is that individual clusters having high similarity across ontology entities are with high probability of the same domain. This hypothesis is backed up with observed patterns of collocated terms within the same domain, and consequently different domains will have different collocation pattern of terms. However, the similarity of clusters depends a lot on the quality of the ontology, especially on semantic distance between the different entities. The result of this step is a Domain Relevance Measure (DRM) score for each cluster of an entity. The relations of each entity are given different weighting according to Definition 4.

Definition 4: Domain Relevance Measure (DRM)

Let $S = \{S1, S2, S3\} = \{parents[e], children[e], other[e]\}$, $c_i \in \{clusters[e]\}$, and $c_k \in \{clusters[S_j]\}$

$$DRM(e, c_i) = \sum_{j=1}^3 \frac{1}{n_j} \sum_{k=1}^{n_j} w_j S_j sim(c_i, c_k) \quad (4)$$

where $DRM(e, c_i)$ is the Domain Relevance Measure for entity e and cluster c_i of e . w_j is a weight factor set to a default value of 1, and S_j is either 1 if S_j is true or 0 if S_j is false. Further, n_j is the number of clusters of each neighbouring entity defined in S .

Note that if a neighbouring entity already has an FV assigned (Step 3.2) then that FV is used in the calculation of the DRM score in contrast to do comparing with all the CLFVs of the neighbouring entities.

Step 3.2: Construct feature vector

The cluster with the highest DRM score, calculated in Step 2.1, is selected for each entity. The step of creating the final FV for the selected cluster can either be based on the already created CLFV of that cluster (Step 2.4) or a deeper analysis of the documents of the selected cluster can be done. In the experiments described in Section 4.2, the CLFVs are used.

The FV, created in this step, are next used to identify the most prominent clusters of the neighbouring entities in Step 3.1. Recall from Step 1, where we argued the importance of finding the most representative entities of the ontology. This ranked list of entities can potentially improve the process of identifying the most relevant clusters w.r.t. the ontology. Using those selected FVs has several advantages. First, the algorithm becomes more efficient while it does not have to calculate the similarity with all the clusters of the already processed entities but only their associated FVs. Secondly, it is assumed that the associated FVs have high relevance to the ontology and therefore good candidates to identify the best CLFV candidate of the neighbouring entities. However, a potential problem with this approach is the drifting of focus (i.e., an erroneous candidate cluster is selected which next is used to find the most prominent candidate cluster of a neighbouring entity, and so forth). Alternatively, all the clusters of an entity are compared with all the clusters of the neighbouring entities independent of the selected clusters of the

neighbours (the method used and described in [16]), which will always be the case of the first concept to be processed.

3.3 Feature Vector Construction Example

In this section, a small example is presented to illustrate the steps of the FVC algorithm described in Section 3.2.

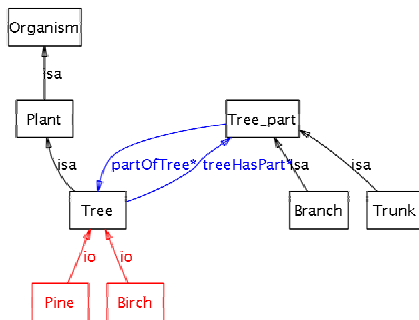


Figure 3. A fragment of ontology describing a tree

Step 1: Rank entities

The example ontology is presented in Figure 3. A ranked list with default weighting values is shown in Table 1.

Table 1. Ranking of the entities found in the example ontology presented in Figure 3.

Concepts	CCM	CDM	CCS
Tree	2.0	4.0	5.0
Tree_part	1.0	3.0	3.5
Plant	2.0	2.0	3.0
Birch	1.0	2.0	2.5
Branch	1.0	2.0	2.5
Pine	1.0	2.0	2.5
Trunk	1.0	2.0	2.5
Organism	1.0	1.0	1.5

Step 2.1: Compose entity query

For the entity Plant seen in Figure 3, the search query will be as follows when *parent* and *child* entities are included:

```
plant organism tree
```

Step 2.2: Entity based search

A search based on the query created in Step 2.1 is performed. The top ranked document by Yahoo!, as of 11th of May 2008, based on this query where a page titled "Green plants" from the "Tree of Life Web Project" Web site (text fragment shown in Figure 4).

"Green plants as defined here includes a broad assemblage of photosynthetic organisms that all contain chlorophylls a and b, store their photosynthetic products as starch inside the double-membrane-bounded chloroplasts in which it is produced, and have cell walls made of cellulose Raven et al., 199. In this group are several thousand species of what are classically considered green algae, plus several hundred thousand land plants."
plant = {(broad assemblage, 1)(contain chlorophylls, 1)(green, 1)(photosynthetic organisms, 1)(plants, 1)}

Figure 4. A text fragment is shown at the top and a set of corresponding extracted key-phrases is seen at the bottom of the figure.

Step 2.3: Contextual key-phrase extraction

For illustration purposes, a small text fragment (Figure 4) is used to illustrate contextual key-phrase extraction

process. Since the text fragment, in this case, is very small, a contextual window of size 10 is used for the query terms. The extracted key-phrases are shown in the bottom of Figure 4.

Step 2.4: Cluster search results

We used the Lingo clustering algorithm from the Carrot API [6]. Twenty-five documents (snippets from Yahoo!™) were used in this case as input to the clustering algorithm. The result was three clusters as shown in Figure 5.

Cluster#1 ={aphis plant, biocontrol organisms, fungi, health plant, home, mycoplasmas, nematodes organism permits, organism, pathogenic bacteria viruses, permits, plant, plant health, plant product, plant protection, soil permits organism permits, usda, viruses}
Cluster#2 ={biology, biology plant, cell, cell biology, cell wall, course, course schedule, disclaimer, email ross, expressions, focus, individual, info homepage email, koning, life course, living, molecular, organism, phys info homepage, plant cell, plant cell wall, plasma membrane, prokaryotic ancestors plant cells, rigid wall, schedule plant, structure cell, study, surrounding, thing, university, variety, wall}
Cluster#3 ={animal plant, animal plant fungus, free encyclopedia, individual, individual animal plant, living, micro-organism, model, model organism, model organism wikipedia, organism, organism wikipedia, popular model, species, specific thaliana, system}

Figure 5. The CLFVs for clusters found for the entity "plant".

Step 3.1: Identify domain relevant clusters

By calculating the similarity with the clusters of the neighbouring entities of Plant, which are Organism (parent entity) and Tree (child entity), we can identify the cluster relevant to this domain. Cluster#1 had the highest score and selected as the candidate cluster.

Step 3.2: Construct feature vector

The final step for the Plant entity is the creation of the FV. In this case, no deeper analysis of the cluster documents were done, consequently the selected CLFV was used as depicted in Figure 6.

Plant ={aphis plant, biocontrol organisms, fungi, health plant, home, mycoplasmas, nematodes organism permits, organism, pathogenic bacteria viruses, permits, plant, plant health, plant product, plant protection, soil permits organism permits, usda, viruses}

Figure 6. The selected cluster and its corresponding FV for the entity plant.

4. EXPERIMENTS

We have conducted a set of experiments (described in Section 4.2) to validate the feature vector construction algorithm presented in Section 3.2. The goal of the experiments is to measure the sensitivity w.r.t. both some of the components of the approach and some heterogeneous ontologies of different granularity (presented in Section 4.1). Consequently, we did not focus on performance issues like processing time, scalability, etc. in this evaluation. We used the Normalized Google Distance (NGD) [8] and three additional measures (presented in Section 4.3) to get a representative value of the FV quality. In Section 5, we present and discuss the results of the experiments.

4.1 Ontologies

FVs' construction is semantics based and heavily relies on ontologies. Consequently, we would like to measure the effect of ontologies of different granularity. We would particularly test the sensitivity of FV quality based on entities processed randomly versus the ordered list of the entities (Step 1 of the algorithm presented in Section 3.2). Three ontologies formalized in OWL were selected:

Animals ontology: A small ontology that classifies some species, does not contain any individuals, and has only hierarchical properties. The ontology was selected to see the effect of applying the approach on a typical *taxonomy*.

Travel ontology: A bit more advanced compared to the Animals ontology by having in addition both individuals and some object properties. This ontology is classified in this work as a *lightweight* ontology.

Wine ontology: Even more advanced than the Travel ontology with more individuals than classes and many relations. This ontology was originally constructed to test reasoning capabilities. Maybe as a result, the ontology contains some entity labels that are not found elsewhere (e.g. the entity *McGuinness* is according to the ontology a winery; however a search with Google provides no results). Consequently, several entities will not be populated with this ontology. This ontology is classified in this work as *advanced*.

The key characteristics of the ontologies are displayed in Table 6 (the ontologies can be accessed at: <http://research.idi.ntnu.no/IIP/ontologies/>).

Table 2. Ontology key characteristics.

Ontology	Classes	Individuals	Properties
Animals	51	0	0
Travel	34	14	6
Wine	82	155	10

We have decided to exclude large or *heavyweight* ontologies in this experiment since we believe that larger ontologies will not provide any significant new insight except of processing time, which is not the focus of this evaluation.

The evaluation has restrictions as follows:

- All OWL object properties are treated as *other* relations.
- Disjointed classes as a feature are ignored since we do not consider siblings in this evaluation.
- The following equality features are ignored: `equivalentClass`, `sameAs`, and `differentFrom`.
- The maximum length of the FVs has been set to 30 (top 30 selected by highest frequency). In earlier experiments, with no restrictions on FV length, the average length was 24.

4.2 Experiments

In this section, we describe the experiments and the motivation behind them.

Table 3. Summary of the experiments conducted.

	BI	Ex#1	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10
Ontology analysis											
with reasoner									X		X
without reasoner	X	X	X	X	X	X	X	X		X	
Entity listing											
with ranking									X		X
without ranking	X	X	X	X	X	X	X	X		X	
Query expansion											
parents		X	X	X						X	X
children			X	X						X	X
others			X	X						X	X
Search results											
nbr of results	30	30	30	30	100	30	30	30	100	100	
Clustering											
document fv					X					X	X
text	X	X	X	X		X	X	X	X		
Domain identification											
parents	X	X	X	X	X	X	X	X	X	X	X
children						X	X			X	X
others							X				

Ontology analysis (Ex#8): A reasoner can extract more neighbouring entities for each entity, which influence the query expansion (Step 2.1, Section 3.2) and the process of identifying the most prominent candidate cluster (Step 3.1, Section 3.2). It is assumed beneficial for an entity to have several neighbours in this process but too many can also be a problem.

Ranking of entities (Ex#8): We test the sensitivity of FV quality based on entities processed randomly versus the ordered list of the entities. However, a potential problem with this approach is the drifting of focus (Step 3.2, Section 3.2). Can drifting of focus affect the FV quality negatively or positively?

Query expansion (Ex#1-3): We test what kind of neighbouring entities (parent, child, other) are optimal to include.

Number of search results (Ex#5): Is 30 search results an optimal number and what implication has it on the FV quality? We test if 100 are more optimal.

Clustering input (Ex#4): The clustering algorithm used is optimized for processing snippets. As a result, it is assumed that using whole documents feature vectors (DFV) will provide better results than using raw full text documents, because the DFVs are contextual and consequently more focused than using the whole documents.

Domain identification (Ex#6, 7): It is expected that comparing neighbouring entities by relation type filtering will have a major effect on the FV quality. Utilizing parents is assumed to have the most positive effect.

Best practice (Ex#9, 10): As the experiment proceeded we started to get some indications of what components and parameters that had a positive effect on the FV quality or not. Consequently, we have combined some of these findings to assess the effect.

4.3 Evaluation Measures

In this section, we will present the evaluation measures used (more details are found in [16]).

Recall that one of the goals of the experiments is to measure the sensitivity of the approach with respect to

some ontologies of different granularity. Consequently, we need to measure the changes to the FVs w.r.t. the ontologies. Four measures were defined. Both Average FV Similarity (AFVS) and Average FV Neighbourhood Similarity (AFVNS) are intrinsic measures indicating the uniqueness and the neighbourhood similarity aspects of the FVs. The Average FV NGD (AFVNGD) is a measure used to find the semantic distance between the entities and their FVs. In addition, a total score (FV Quality Score (FVQS)) is defined being an aggregated score of the above three measures. These scores give a representative value of the FV quality w.r.t. the ontologies.

First, we define the Average FV Similarity (AFVS). AFVS gives an indication of the uniqueness of the FVs and is defined as follows.

Definition 5. Average FV Similarity (AFVS):

$$AFVS(o) = \frac{2}{n^2 - n} \sum_{i=1}^n \sum_{j=i+1}^n sim(fv_i, fv_j) \quad (5)$$

where n is the number of fv s in the ontology o and $sim(fv_i, fv_j)$ is a similarity between the two vectors. A score of zero indicate that all FVs are unique. In general, we would like this score to be as low as possible in order to discriminate the FVs, but this depends a lot on the quality of the ontology.

The Average FV Neighbourhood Similarity (AFVNS) score indicates the degree of overlap with neighbouring entities and is defined as follows.

Definition 6. Average FV Neighbourhood Similarity (AFVNS):

$$AFVNS(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m sim(fv_i, fv_j) \quad (6)$$

where n is the number of FVs in the ontology o and m is the number of neighbouring entities with FVs of entity i with FV_i . In this experiment, we have selected to use all the neighbours of an entity and do not differentiate the neighbours by weighting.

Normalized Google Distance (NGD) [8] was used in the Average FV NGD (AFVNGD) score. The AFVNGD score indicates the semantic distance between the entities and their FVs and is defined as follows.

Definition 7. Average FV NGD (AFVNGD):

$$AFVNGD(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m NGD(fv_i, kp_j) \quad (7)$$

where n is the number of fv s in the ontology o and m is the length of the fv_i and fv_{n_i} is the name of the fv_i , the entity name, and kp_j are the key-phrases of fv_i . Note, if an entity has a parent, then the name of the parent is also included to provide a more specific similarity distance (adapted from [8]). The range of $FvNGD(fv)$ is between 0 and ∞ , where 0 denotes best match (in practice most values are in the range from 0 to 1).

Once AFVS, AFVNS, and AFVNGD are found the total score can be calculated. The total score is an

aggregated score of the three measures. The total FV quality score is defined as follows.

Definition 8. FV Quality Score (FVQS) provides the overall quality of the FVs.

$$FvQS(o) = \alpha(1 - AFVS) + \beta AFVNS + \gamma(1 - AFVNGD) \quad (8)$$

where $\alpha + \beta + \gamma = 1$ are weight factors (defaults are 1/3). The total FV quality score for an ontology will be in the range 0-1, where 1 indicates the best score.

To evaluate the effect of the experiments we needed a baseline (denoted as Bl in Table 3). The baseline was conducted twice, at the beginning and at the end of the experiments, to discount the effect of uncontrollable external changes (e.g., change of ranking by the search engine provider).

5. RESULTS AND ANALYSIS

In this section, the results of the experiments are presented and analysed.

5.1 Results

Table 4 summarises the test results. 10 experiments (Table 3) were conducted on three ontologies resulting in 30 different configurations. The experiments were performed on a standard PC with an Intel™ Pentium processor running Windows™ XP, running Apache Tomcat. Populating the ontologies took more than 13 hours; the most complex ontology, the Wine ontology, took from 16 to 298 minutes to populate. When evaluating the quality of the FVs using NGD, more than 260.000 queries were submitted. The evaluation was conducted in the course of one week.

Table 4. Experimental results.

	Bl#1	Bl#2	Ex#1	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10
Animals ontology												
AFVS	0.019	0.020	0.048	0.040	0.039	0.014	0.017	0.018	0.018	0.021	0.044	0.043
AFVNS	0.154	0.168	0.304	0.404	0.406	0.224	0.210	0.181	0.172	0.173	0.487	0.495
AFVNGDS	0.266	0.255	0.194	0.214	0.215	0.249	0.259	0.260	0.231	0.276	0.198	0.209
FvQS	0.623	0.631	0.687	0.717	0.717	0.654	0.644	0.635	0.641	0.625	0.749	0.748
Travel ontology												
AFVS	0.019	0.023	0.042	0.035	0.034	0.049	0.022	0.026	0.023	0.025	0.043	0.046
AFVNS	0.186	0.147	0.326	0.243	0.272	0.225	0.201	0.136	0.137	0.069	0.343	0.320
AFVNGDS	0.253	0.253	0.227	0.231	0.233	0.229	0.241	0.236	0.234	0.251	0.237	0.225
FvQS	0.638	0.624	0.686	0.659	0.668	0.649	0.646	0.625	0.627	0.598	0.687	0.683
Wine ontology												
AFVS	0.040	0.041	0.079	0.075	0.073	0.099	0.045	0.045	0.042	0.064	0.101	0.134
AFVNS	0.286	0.286	0.412	0.403	0.412	0.446	0.386	0.308	0.311	0.240	0.553	0.442
AFVNGDS	0.163	0.180	0.149	0.150	0.149	0.196	0.184	0.156	0.180	0.168	0.174	0.169
FvQS	0.694	0.688	0.728	0.726	0.730	0.717	0.719	0.703	0.696	0.669	0.759	0.713

5.2 Analysis

An overview of the experiments and results deviation from the baseline (in percents) is shown in Figure 7. Since we used Bl#1 as the baseline the values for this experiment is set to zero. We had limited control of external changes (like change of parameters) that might affect the search results (like different ranking), since we were depended on performance of a commercial search engine. Therefore, the first test was also repeated last. The test is denoted Bl#2 and serves as the deviation and therefore subtracted from the results shown in Figure 7. Figure 8 presents the error rates. The error rate is the ratio of entities populated versus not populated. For most of the experiments, a slight decrease in errors is observed. However, for Ex#8 we

observe an increase in errors, which can be caused by the drifting problem (Step 3.2, Section 3.2). Next, we discuss results of the experiments.

Ontology analysis (Ex#8): Ex#8 scored on average slightly worse than the baseline. A surprise was the decrease in score for the Travel and Wine ontologies. A reason for this might be that the additional relations provided by reasoning are not fully utilized, which we test in Ex#10. However, if we look at the error rates in Figure 8 we see a decrease in errors for the Travel and Wine ontologies, which can indicate that more relations provided by the reasoner helps to populate more entities.

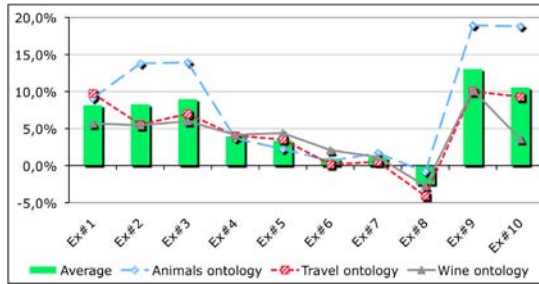


Figure 7. Experimental analysis.

Ranking of entities (Ex#8): Another surprise was the slight decrease in the FV quality score in Ex#8 when ranking the entities. This is probably because no entities (other than the parent entities) were used when identifying the most prominent cluster candidates. However, in Ex#10 children entities were used but still we had the same tendency, a lower score than the comparable test Ex#9. The error rates in Figure 8 had increases too. Combination of these observations indicates the drifting problem described in Step 3.2 (Section 3.2).

Query expansion (Ex#1-3): Ex#3 provided on average the best results, and the best results for the Animals and the Wine ontologies. However, for the Travel ontology Ex#3 provided the fourth best results. It was assumed that Ex#1 on average would provide the best results but it turned out to be the third best. If we look at both the standard deviation and mean results, then Ex#1 yields the best results (Table 4 and Figure 7). This could indicate that independent of the quality of the ontology, Ex#1 would be the best choice.

Number of search results (Ex#5): Here we tested if the number of search results retrieved and processed would affect the FV quality, which is the case with 3.4%. More clusters are more expensive to compute. In Ex#1, the Animals, Travel and Wine ontologies took respectively 3, 3, and 16 minutes to process, while processing of Ex#5 was on average 3 times longer. In this experiment, we have not tried to find the optimal number of results to process, but just by looking at the increase of the FV quality from 30 to 100 results indicates that 100 is a better candidate in this test w.r.t. both the FV quality.

Clustering input (Ex#4): In Ex#4 we tested if it is more beneficial to use document FVs (DFV) as input to the

clustering algorithms or snippets. Ex#4 showed some improvement of using DFVs compared to the baseline.

Domain identification (Ex#6, 7): Not surprisingly, we got more or less the same results as for the query expansion experiments (Ex#1-3) where using *parents*, *children*, and *other* neighbouring entities provided the best results (Ex#7).

Best practice (Ex#9, 10): These experiments were conducted to test the combination of some of the best results from the other experiments. Both Ex#9 and Ex#10 performed considerably better than the other experiments with an increase of 13.1% and 10.6% respectively. The score for the Wine ontology in Ex#10 performed considerably worse than for Ex#9; this is probably due to the ranking problem as described for Ex#8. However, the error rates for Ex#10 (Figure 8) are very low indicating that additional relations provided by a reasoner have a positive effect on populating more entities.

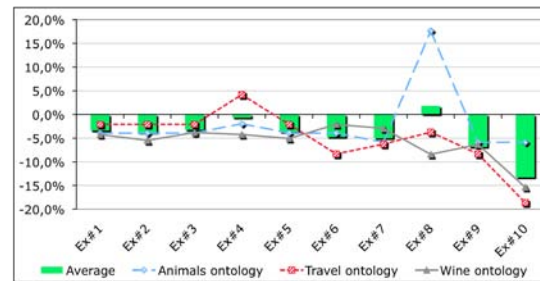


Figure 8. Error rate analysis.

5.3 Key findings

Based on the findings in the conducted experiments we conclude the following:

Taxonomy kind of ontologies (e.g. Animals):

- *Query expansion:* Usage of parent entities when expanding the query provides the best results.
- *Clustering input:* Using full text documents in combination with extraction of the most relevant key-phrases seems to provide the best positive effect on the FV quality.
- *Domain identification:* Including the parents, children, and other related entities seem to provide the best results when identifying the most prominent cluster candidates.

Lightweight ontologies (e.g. Travel):

- *Query expansion:* Usage of parent entities in combination with scope keywords provides the best results.
- *Clustering input:* Using full text documents in combination with extraction of the most relevant key-phrases seems to provide the best positive effect on the FV quality.
- *Domain identification:* Including the parents and other related entities seem to provide the best results.

More advanced ontologies (e.g. Wine):

- *Query expansion*: Usage of parents, children, and other related entities are recommended to provide the best results.
- *Clustering input*: No recommendation. Further research needed, since the Wine ontology used in these experiments is probably not representative.
- *Domain identification*: Including parents and other related entities seem to provide best results for advanced ontologies.

Moreover, the most important component with respect to the FV quality in general is the query expansion component. The parent entities are the most important neighbouring entities. Therefore, FV construction for taxonomy type ontologies (e.g., Animals) is most sensitive to different techniques (see Table 4), while advanced and rich ontologies as Wine are least sensitive. This indicates that FV construction process needs to be tuned mostly for taxonomy type of ontologies, whereas rich ontologies have a lot of knowledge. The knowledge contained in the ontologies provides enough good substance for FV construction, i.e. the construction process is not so sensitive to the processing techniques, though quality still can be improved.

Further, we found that ranking of entities had a negative effect on the FV quality when compared with the algorithm without ranking [16]. Moreover, surprisingly, the total processing time increased mainly because of the complexity of the ranking algorithm. From another hand, the domain identification process took less time because fewer comparisons needed to be done.

6. CONCLUSION

In this study, we have described and evaluated an unsupervised approach to feature vector construction. The proposal is based on a non-supervised solution that is applicable to any ontology as long as there is some correlation between the ontology and the text corpus. We have described the process of associating each entity of an ontology with a semantically enriched FV.

In evaluation we have investigated the aspects of the components w.r.t. both the FV quality and the ontologies used. Ontologies of different granularity have been used and 30 different configurations of experiment have been conducted. The ontologies have been categorised based on key characteristics and trends investigated with regards to the categories revealing that the approach is most sensitive to taxonomy kind of ontologies. Furthermore, ranking of entities neither enhances the FV quality nor speeds up the process.

However, we need to scale up evaluation of the approach with more ontologies. That is one of the main future tasks.

7. ACKNOWLEDGMENTS

This research work is funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30. In addition, we would like to thank Jon Atle Gulla (NTNU), Per Gunnar Auran (Yahoo!), and Robert Engels (ESIS) for their support and help.

8. REFERENCES

- [1] Agirre, E., Ansa, O., Hovy, E.H., Martinez, D.: Enriching very large ontologies using the WWW. ECAI Workshop on Ontology Learning, Vol. 31. CEUR-WS.org (2000)
- [2] Alani, H., Brewster, C., Shadbolt, N. Ranking Ontologies with AKTiveRank. In *The Semantic Web - ISWC 2006*, LNCS 4273 (2006) 1-15.
- [3] Bergamaschi, S., Bouquet, P., Giazomuzzi, D., Guerra, F., Po, L., Vincini, M.: An Incremental Method for the Lexical Annotation of Domain Ontologies. *Int. J. on Semantic Web and Information Systems* 3(3) (2007) 57-80
- [4] Bouquet, P., Serafini, L., Zanobini, S.: Semantic Coordination: A New Approach and an Application. *The SemanticWeb - ISWC 2003*, LNCS 2870, (2003) 130-145
- [5] Bry, F., Koch, C., Furche, T., Schaffert, S., Badea, L., Berger, S.: Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages. *Int. J. on Semantic Web and Information Systems* 1(2) (2005) 1-21
- [6] Carrot2: Carrot2 an Open Source Search Results Clustering Engine. (2009)
- [7] Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. *IEEE TKDE* 19(2) (2007) 261-272
- [8] Cilibrasi, R., Vitanyi, P. The Google Similarity Distance. *IEEE TKDE* 19(3) (2007) 370-383.
- [9] Crestani, F. Application of Spreading Activation Techniques in Information Retrieval. *Artificial Intelligence Review*, 11 (6). 453-482.
- [10] Gabrilovich, E., Markovitch, S.: *Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization*. *J. Mach. Learn. Res.* 8 (2007) 2297-2345
- [11] Panagis, Y., Sakkopoulos, E., Garofalakis, J., Tsakalidis, A. Optimisation mechanism for web search results using topic knowledge. *Int. J. Knowledge and Learning* 2(1/2) (2006) 140-153
- [12] Strasunskas, D., Tomassen, S.L. The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation. *Int. J. Knowledge and Learning*, 4 (4). 398-414.

- [13] Su, X., Gulla, J.A.: An information retrieval approach to ontology mapping. *Data & Knowledge Engineering* 58 (2006) 47-69
- [14] Suomela, S., Kekalainen, J.: Ontology as a search-tool: A study of real user's query formulation with and without conceptual support. In: Proceedings of ECIR'2005. LNCS 3408, Springer-Verlag (2005) 315-329
- [15] Tomassen, S.L., Strasunskas, D. Construction of Ontology based Semantic-Linguistic Feature Vectors for Searching: the Process and Effect. In *IEEE/WIC/ACM Int. Conf. on Web Int. and Int. Agent Technology (WI-IAT '09)*, IEEE Computer Society, Milano, Italy (2009) 133-138.
- [16] Tomassen, S.L., Strasunskas, D. Semantic-Linguistic Feature Vectors for Search: unsupervised construction and experimental validation. In *ASWC 2009*, Springer-Verlag, Shanghai, China (2009).
- [17] Yang, H-C. A method for automatic construction of learning contents in semantic web by a text mining approach. *Int. J. Knowledge and Learning* 2(1/2), (2006) 89-105

Errata

Table 1 uses the old terminology and should be replaced by the following table with the new terminology.

Entities	EBM	EDM	ECS
Tree	2,0	4,0	5,0
Tree_part	1,0	3,0	3,5
Plant	2,0	2,0	3,0
Birch	1,0	2,0	2,5
Branch	1,0	2,0	2,5
Pine	1,0	2,0	2,5
Trunk	1,0	2,0	2,5
Organism	1,0	1,0	1,5

P4: Measuring intrinsic quality of semantic search based on Feature Vectors

Publication details

Tomassen, S.L. & Strasunskas, D. (2010) Measuring intrinsic quality of semantic search based on Feature Vectors. *Int. J. Metadata, Semantics and Ontologies*, 5(2), pp. 120-133.

Measuring intrinsic quality of semantic search based on feature vectors

Stein L. Tomassen*

Dept. of Computer and Information Science
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
E-mail: steint@idi.ntnu.no
* Corresponding author

Darijus Strasunskas

Dept. of Industrial Economics and Technology Management
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
E-mail: darijuss@gmail.com

Abstract: Search is probably the most frequent activity on the Web. Yet it is not effortless, mainly due to heterogeneous information resources. Semantic search is a means to tackle the problem of ambiguity. In this paper, we analyse a process of constructing semantic-linguistic Feature Vectors (FV) used in our semantic search approach. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. Since FVs are central building blocks of the approach, we investigate the quality of FVs. We take a closer look at the process of FV construction and the impact of chosen techniques on the quality of FVs. We report on a set of laboratory experiments and analyse aspects affecting the FV quality and the FV construction error rates.

Keywords: Semantic search; FVC; feature vector construction; evaluation; ontology.

1 Introduction

Nowadays, the Web is becoming one of the dominant information sources for learning and acquiring new knowledge. However, finding relevant information is still a huge challenge. The Semantic Web (SW) (Berners-Lee et al., 2001) is believed to be the successor of the current Web and provides a means to tackle some of these issues of the current Web (van Harmelen, 2006). Ontologies are the building blocks of the SW and are used to encode knowledge about the domain of interest by standardising and disambiguating domain terminology. As a result, much research has been devoted to the improvement of search performance using ontologies (e.g., Nagypal, 2005; Jiang & Tan, 2006; Castells et al., 2007; Formica et al., 2008; Solskinnsbakk & Gulla, 2008).

Our approach to the semantic search is based on Feature Vectors (FV), which are created in order to "bridge" or connect the standardised domain terminology (encoded in an ontology) to the actual terminology ("slang") used on the Web.

The underlying idea is that a FV reflects both the semantic and linguistic neighbourhoods of a particular entity. The semantic neighbourhood is computed based on related entities and direct properties specified in an ontology, while the linguistic neighbourhood is based on the collocation of terms in a domain specific corpus. Therefore, every entity (classes and individuals) of the ontologies is associated with a FV to tailor it to the specific terminology of the text corpus. We give an overall description of how FVs are constructed. However, in this paper we do not go into details of this algorithm (details can be found in [Tomassen & Strasunskas, 2009]). The extrinsic quality of FV (i.e., its effect on search performance) has been investigated in (Strasunskas & Tomassen, 2008). There we reported an improvement of the search by more than 10%, on average. Real users have conducted the experiment. However, because of variances in the results (partly explained by the diversity of users), we needed to assess the intrinsic quality of FV by evaluating the Feature Vector Construction (FVC) process. Therefore, in this paper we focus on the

aspects of the components of this construction algorithm that affect the intrinsic FV quality. In the evaluation, we analyse the effect of alternative techniques (that are used to construct FVs) on the FVs. Finally, we predict the potential search improvements based on the findings from these experiments.

Our approach to the construction of FVs is based on a non-supervised solution that is applicable to any ontology and text corpus as long as there is some correlation between them. However, the approach is not dependent on a collection of only relevant documents. On the contrary, the main advantage of the approach is that a diverse corpus, like the Web can be used. Our approach is capable of disambiguating word sense by utilising the relationships between the entities. Nevertheless, the FV quality will be highly dependent on both the quality of the ontology and correlation of terminologies in the ontology and the text corpus.

1.1 Paper contribution

Feature vectors are widely used in many different applications like ontology alignment, ontology mapping, semantic search, and ontological filtering, etc. (Su & Gulla, 2006; Lopez et al., 2006; Formica et al., 2008; Solskinnsbakk & Gulla, 2008). However, most evaluate FVs as a black box, i.e. evaluating the end-result of the

system. Given the number of FV applications in different areas, this paper provides useful intrinsic insights on how the process of FV construction can be evaluated and the FV quality assessed. Therefore, the main contribution of this paper is as an evaluation method and to describe lessons learnt. The evaluation focuses on the FVs' sensitivity to the construction process and alternative techniques used in the FV creation process. Furthermore, the effect of the ontology's granularity on FV quality has been investigated.

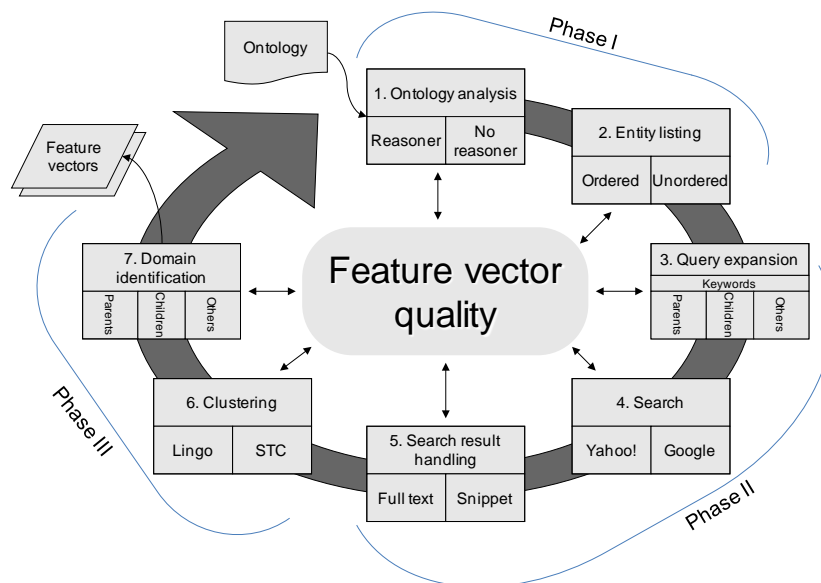
1.2 Paper overview

This paper is organised as follows: In section 2, the feature vector construction process is detailed. In section 3, the experiments are described and the evaluation measures are defined. In section 4, the results are presented and analysed. Section 5 is designated to an overview of related work and the positioning of our approach. Finally, in section 6, we conclude the paper.

2 Feature vector construction

In this section, we present the feature vector construction (FVC) process (details are found in [Tomassen & Strasunskas, 2009]). First, we provide the definition of FVs and then we elaborate on each of the FVC steps.

Figure 1 An overview of the FV construction process. The input to the process is an ontology and the output is a set of associated FVs for each entity of the input ontology.



2.1 Definition of feature vectors

The development of the approach is inspired by a linguistics method for describing the meaning of objects - the semiotic triangle by Ogden and Richards (1930). In our approach, a FV "connects" a concept (entity) to a document collection, i.e., the FV is tailored to the specific terminology used in a particular collection of the documents. Therefore, a FV constitutes a rich representation of the entities and is related to the actual terminology used in the text corpus. Correspondingly, a FV of an entity e is represented as a two-tuple as follows:

Definition 1: Feature Vector (FV)

$$FV_e = \langle S_e, L_e \rangle \mid S_e \in O_d, L_e \in D_d$$

$$S_e = (e_i, DR_{e_i})$$

$$DR_{e_i} = Parents_{e_i} \cup Children_{e_i} \cup Others_{e_i} = \{ \langle e_i, e_k \rangle \} \subseteq E \times E$$

$$L_e = collocated(S_{e_i}, L_{e_{ind}})$$

where S_e is a semantic enrichment part of FV_e that represents a set of neighbourhood entities and properties in an ontology O of a domain d . L_e is a linguistic enrichment of an entity that is a set of terms (from document collection D of a particular domain d) with a significant proximity to an entity and its semantic neighbourhood. A *parent* entity of a class is defined to be its super-class, while a parent of an individual is the class of which the individual is an instance or part. A *child* entity of a class is defined to be its sub-class or individual, the latter if it does not have a sub-class. Finally, *other* neighbouring entities are any other related entities.

2.2 Feature vector construction steps

The FVC process (depicted in Figure 1) is composed of three phases. The first phase (Steps 1 and 2) includes preparing the ontology for further processing. First, the ontology is analysed to find the entities and the relationships among them. Next, the ontology entities are listed. The main aim of second phase (Steps 3 to 5) is to find candidate documents that are potentially relevant to the entities. By submitting an entity-based query to a Web search engine, we get a set of potentially relevant documents. The last phase (Steps 6 and 7) include grouping documents and identifying the most relevant groups w.r.t. the ontology. The documents retrieved by the search engine typically will represent several domains that can be found by clustering. However, it is not obvious which of

these candidate clusters (domains) is most relevant to the ontology. Consequently, the main aim of this last phase is to identify the most relevant candidate cluster w.r.t. the entities and hence the ontology. Finally, a FV for each entity is created based on the most prominent candidate cluster for each entity. The result of these steps is a list of entities with corresponding FVs that consist of terms associated with both the entities and the domain terminology.

Step 1: Ontology analysis

The first step includes loading an ontology and analysing it to find the relationships among the entities. The ontologies are expressed in OWL (W3C, 2004). When loading an ontology, typically only some of the relationships are found. However, a semantic reasoner can be utilised, like the Pellet OWL Reasoner (Sirin et al., 2007), to extract all the relationships among the entities. Though, the question is, does this extra knowledge provided by a reasoner increase the quality of the FVs?

Using a reasoner will consequently affect the number of relations for each entity that will be available. However, it can also affect the number of entities being available. For example, if two classes are sub classes of `owl:Thing` and `equivalent`, then all their properties and sub hierarchical structure should be equal. These "additional" relations are found by a semantic reasoner.

Step 2: Entity listing

After analysing the ontology, the entities are either sorted or unsorted. The entities can be ordered according to considered importance (centrality) w.r.t. the ontology or unordered (random). It is assumed that an ordered list can positively improve the selection of the most relevant candidate clusters (in Step 7). However, the question is, does an ordered list of entities improve the identification of relevant candidate clusters versus and unordered list?

A ranked list of entities is assumed to positively improve the selection of the most relevant candidate cluster (the ranking algorithm is based on *AktiveRank* by Alani et al. [2006] and is thoroughly described in [Tomassen & Strasunskas, 2009]). An underlying assumption is that more information is available for the most central entities (they are the most semantically rich by having the largest number of relations and by being central to other entities) and consequently they are better candidates to distinguish relevant

candidate clusters. The most prominent cluster candidates are later used to identify new candidate clusters, and so on. However, a potential problem with this approach is the drifting of focus (i.e., an erroneous candidate cluster is selected which next is used to find the most prominent candidate cluster of a neighbouring entity, and so forth).

An alternative to a ranked list of the entities is a random list. Instead of using the most prominent cluster candidates to identify new cluster candidates, all the candidate clusters of the neighbouring entities are used. By always comparing all of the candidate clusters of the neighbouring entities, the potential drifting problem, described above, can be avoided at the cost of longer processing time (i.e. more similarity calculations).

Step 3: Query expansion

In this step, a search query is prepared for each entity while the actual search is performed in Step 4. The query is based on the entity name and optionally expanded with selected neighbouring entities and/or keywords. The motivation behind expanding the initial query with neighbouring entities is to create a query that reflects both the ontology and the relationship of each entity to other neighbouring entities. The question at this stage is, what kind of neighbouring entities is optimal to include? Do keywords provide better FVs?

The neighbouring entities are grouped according to their relation type (described in Definition 1). We have also added an option to include keywords that typically represent the ontology as a whole. Larger ontologies tend to include several minor domains. E.g., the Wine ontology used in the experiments includes the Food ontology. A user, using this ontology and searching using the 'Lobster' entity, would expect to get lobster in relation to wine results since 'Lobster' is part of the Wine domain (ontology). Therefore, using keywords will create FVs that are more homogeneous and hence can be beneficial when creating FVs that are more true to the domain defined by the ontology as a whole (Gligorov et al., 2007). Omitting keywords would create FVs that are more unique and hence truer to the local variances in the ontology and not necessarily to the ontology as a whole.

Step 4: Entity based search

The queries created in the previous step are used in this step to retrieve candidate documents for each

entity. Any search engine can be used, but Yahoo!TM and GoogleTM are used in the experiments. The question is, does a change of search engine affect the quality of the FV?

Step 5: Search result handling

The retrieved documents, from Step 4, function as input to this step. Either full text documents or the snippets (document summaries from the search engines) can be used. Further, the full text documents or the snippets can either be processed by creating document feature vectors (DFV) or keeping the texts in their raw form (e.g., the text without HTML tags). However, the question is, does the FV quality improve by using full text documents compared to using only snippets?

The snippets from both of the engines are comparable in length, on average about 140 characters. The DFVs contain key-phrases that are extracted from the documents/snippets. However, only those key-phrases that are within a so-called "contextual window" are extracted. A contextual window is a frame of a specified size (e.g., 100 characters) surrounding a key-phrase. Finally, only the most prominent candidate key-phrases are selected and stored in the DFVs.

Step 6: Clustering

In order to identify (discriminate) different domains within the documents found for each entity, clustering techniques are used. In the experiments, we use Lingo and STC, both part of the Carrot² framework (Carrot2, 2009). Even though the FVC algorithm is not designed for any particular clustering algorithm, we need to test whether a change of algorithm has any major impact on the feature vector quality.

At this stage of the process, the ontology entities are treated as ordinary words and can consequently be part of many different domains (e.g., 'Jaguar' can both be an animal and a car brand). Clustering is done with the purpose of indicating different domains. The result of this step is a set of clusters for each entity. In addition, for each cluster a cluster feature vector (CLFV) is associated that is a product of all the DFVs of a cluster.

Step 7: Identifying domain relevant clusters

A problem at this stage is to identify the most relevant clusters, made in Step 6, w.r.t. the ontology. Therefore, we compute the similarity between the CLFVs of an entity with the CLFVs of the selected neighbouring entities. In order to find

the most prominent cluster, an entity must have at least one neighbour, otherwise this check would fail. The neighbouring entities are grouped according to their relation type, as in Step 3. The question is, what kinds of neighbouring entities contribute most to the FV's quality?

Commonality (i.e. high similarity) here identifies the document sets (clusters) being relevant to the domain of interest. An assumption is that individual clusters having high similarity across ontology entities have a high probability in the same domain. This hypothesis is backed up with observed patterns of collocated terms within the same domain, and consequently different domains will have a different collocation pattern of terms (more details are found in [Tomassen & Strasunskas, 2009]). However, the similarity of clusters depends a lot on the quality of the ontology, especially on the semantic distance between entities.

The result of this final step is a FV associated for each of the ontology entities.

3 Experiments

We have conducted a set of experiments to validate the FVC algorithm proposed in Section 2. The goal of the experiments is to measure the sensitivity w.r.t. both the components of the FVC algorithm and the ontologies of different granularity. Further, we propose using Normalised Google Distance (NGD) (described in Section 3.3) and two additional measures to get a representative value of the FV quality. In this section, we present the ontologies used and the tests conducted, and analytically evaluate the approach. In the next two sections, we present and discuss the results of the experiments.

3.1 Ontologies

Ontologies of different granularity were used to measure their effects on the algorithm. We chose three ontologies that also were used in our earlier experiment on search performance (Strasunskas & Tomassen, 2008). All the ontologies are formalised in OWL and can be found at: <http://research.idi.ntnu.no/IIP/ontologies/>. Next, short descriptions of the ontologies are provided:

- The Animals ontology is a small ontology that classifies some species, does not contain any individuals, and has only hierarchical properties. The ontology was selected to see

the effect of applying the approach on a typical *taxonomy*.

- The Travel ontology is more advanced compared to the Animals ontology by having individuals and some object properties. As a result, more relationships among the entities are available. The ontology is classified in this work as a *lightweight* ontology.
- The Wine ontology is more advanced than the Travel ontology with more individuals and relations. This ontology was originally constructed to test reasoning capabilities. Perhaps, as a result, the ontology contains some entity labels that typically are not found elsewhere (e.g. the entity "McGuinnesso" is according to the ontology, a winery; however, a search using Google™ provides no results of such a winery). Consequently, several entities will not be populated with this ontology. The ontology is, in this work, classified as *advanced* and can, to some extent, indicate the robustness of this approach.

We have decided to exclude *heavyweight* ontologies (i.e. ontologies with several thousand entities) in this experiment since we believe that larger ontologies will not provide any significantly new insight except that of processing time, which is not the focus of this evaluation.

Table 1 Ontology key characteristics.

<i>Ontology</i>	<i>Classes</i>	<i>Individuals</i>	<i>Properties</i>
	<i>n/r</i>	<i>n/r</i>	<i>n/r</i>
Animals	51/51	0/0	0/0
Travel	34/33	14/14	6/6
Wine	82/137	155/194	10/10

n=no reasoner, r=reasoner

The key characteristics of the ontologies are displayed in Table 1. The number of classes, individuals, and properties of an ontology are only relevant in the degree to which they are used. Therefore, we have selected to present different views of these characteristics. The numbers of *classes*, *individuals*, and *properties* are divided into two different views where *n* is the number when no reasoner was used and *r* is the number when a reasoner was used. For instance, the Travel ontology is seen as having 34 classes without using a reasoner while 33 when using a reasoner. The Travel ontology contains a 'Safari' class that is a sub-class of both the 'Adventure' class and the 'Sightseeing' class. Further, when using a reasoner

the representation of the 'Safari' entity is not OWL DL compliant and hence omitted, which results in 33 versus 34 classes.

3.2 Experiments

In this section, we describe the experiments and the motivation behind them. The conducted experiments are summarised in Table 2. There we see minor changes among the experiments to isolate their effect on the FV quality. Next, we briefly describe each of the experiments.

- *Ontology analysis (Ex#22, 24):* A reasoner can extract more neighbouring entities for each entity (see Table 1), which influence the query expansion (Step 3, Section 2) and the process of identifying the most prominent candidate cluster (Step 7, Section 2). It is assumed beneficial for an entity to have several neighbours in this process but too many can also be a problem. Consequently,

we would like to test the effect of utilising more knowledge from the ontologies.

- *Ranking of entities (Ex#23, 24):* We test the sensitivity of FV quality based on entities processed randomly versus the ordered list of the entities.
- *Query expansion - neighbours (Ex#2-8):* We test what kind of neighbouring entities (parent, child, other) are optimal to include.
- *Query expansion - keywords (Ex#28, 29):* By populating an ontology with global keywords (manually selected) it is expected that all the FVs will have higher similarity and be less unique compared to omitting the global keywords.
- *Search engine (Ex#13, 25)* We do not expect any major difference in FV quality when using either Yahoo!™ or Google™.

Table 2 Summary of the experiments conducted.

	BI	Ex#1	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10	Ex#11	Ex#12	Ex#13	Ex#14	Ex#15	Ex#16	Ex#17	Ex#18	Ex#19	Ex#20	Ex#21	Ex#22	Ex#23	Ex#24	Ex#25	Ex#26	Ex#27	Ex#28	Ex#29		
1. Ontology analysis																																
with reasoner																									X	X			X			
without reasoner	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
2. Entity listing																																
with ranking																									X	X			X			
without ranking	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
3. Query expansion																																
<i>neighbors</i>																																
parents			X			X	X		X																				X	X		
children				X		X		X	X																			X	X			
others					X		X	X	X																			X	X			
keywords		X ¹																											X	X		
4. Search engine																																
Yahoo!	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
Google													X													X						
5. Search results																																
<i>content</i>																																
snippet	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
full text										X	X																					
nbr of results	30	25 ²	30	30	30	30	30	30	30	30	30	30	30	100	200	30	30	30	30	30	30	30	30	30	30	60	100	100	30	30		
6. Clustering																																
<i>input</i>																																
document fv											X	X																X	X			
text	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
<i>algorithm</i>																																
Lingo	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
STC													X																			
7. Domain identification																																
<i>neighbors</i>																																
parents	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X		X	X	X	X	X	X	X	X	X		
children	X																	X	X		X	X						X	X			
others	X																		X	X	X											

¹ Animals ontology: 'animals'; Travel ontology: 'travel'; Wine ontology: 'wine'

² 25 search results were originally used in an experiment conducted in 2008

- *Number of search results (Ex#14, 15)* Thirty search results has been set as the baseline. Is this an optimal number and what implication does it have on the FV quality? We test if 100 or even 200 is more optimal.

We expect that more search results will have a positive effect on the FV quality.

- *Content (Ex#9):* It is expected that using full text documents will provide better FV quality than using snippets.

- *Clustering - input (Ex#10, 11)*: The clustering algorithms used are optimised for processing snippets. As a result, it is assumed that using DFVs will produce better candidates than using raw full text documents. However, for snippets, it might be better to use the raw text rather than creating DFVs since snippets are short in length.
- *Clustering - algorithms (Ex#12)*: We test if there are any big differences in the FV quality by using either the Lingo or STC clustering algorithms (Carrot2, 2009).
- *Domain identification (Ex#16-21)*: It is expected that comparing neighbouring entities by relation type filtering will have a major effect on the FV quality. Utilising parents is assumed to have the most positive effect.
- *Best practice (Ex#26, 27)*: As the experiment proceeded, we started to get some indications of which components and parameters had a positive effect on the FV quality. Consequently, we would also like to test if a combination of these findings would yield the same positive effect or not. Therefore, we combined some of these findings in two tests to assess the effect.

To evaluate the effect of the experiments we needed a baseline (denoted as BI in Table 2). The baseline was conducted twice, at the beginning and at the end of the experiments, to discount the effect of uncontrollable external changes (e.g., change of ranking by the search engine provider). To measure this drifting effect, Definition 2 was used. For the domain identification component (Step 7), we selected to use *parent* entities for comparison since it must be compared with at least one neighbouring entity (see Step 7 in Section 2 for details). The experiments were done over a period of one week.

Definition 2 *Web Drift Effect (short term)* (WDE_{ST})

$$WDE_{ST} = |BI\#1 - BI\#2|$$

In addition, we would like to test what effect a time span of one year would have on the quality of the FVs. An experiment was conducted one year prior to the experiments conducted in this paper where real users evaluated the effect of the FVs in a search application (Strasunskas & Tomassen, 2008). Consequently, we had the opportunity to

compare newly populated FVs, with the same parameters, with the one-year-old FVs. This experiment (denoted as Ex#1 in Table 2) was conducted to observe potential content drifting (Definition 3).

Definition 3 *Web Drift Effect (long term)* (WDE_{LT})

$$WDE_{LT} = |Ex\#1a - Ex\#1b|$$

3.3 Evaluation measures

Beauty is in the eye of the beholder. In (Strasunskas & Tomassen, 2008) and (Tomassen & Strasunskas, 2009) end-users were used to assess the performance of our approach to semantic search. Therefore, FV quality was measured indirectly. In this paper, we directly evaluate quality of FVs relative to the ontologies used on the Web. Ideally, Text Retrieval Conference (TREC) corpus would be used but we experienced the same problems as d'Aquin et al. (2008) in finding good ontologies that covered TREC documents and queries. Therefore, we proposed the following intrinsic and extrinsic measures to evaluate the quality of the FVs. The Average Fv Similarity (AFvS) and the Average Fv Neighbourhood Similarity (AFvNS) are both intrinsic measures that indicate the FV quality w.r.t. the ontology (the latter assesses the semantic neighbourhood of the entities). Further, the Average Fv NGD (AFvNGD) is an extrinsic measure that indicates the FV quality w.r.t. the Web. Finally, the Fv Quality Score (FvQS) aggregates the overall FV quality score. Next, the different scores used to measure the FV quality are defined.

The Average Fv Similarity (AFvS) gives an indication of the uniqueness of the FVs and is defined as follows:

Definition 4 *Average Fv Similarity (AFvS)*

$$AFvS(o) = \frac{2}{n^2 - n} \sum_{i=1}^n \sum_{j=i+1}^n sim(fv_i, fv_j)$$

where n is the number of fvs in the ontology o and $sim(fv_i, fv_j)$ is the traditional cosine similarity measure (Baeza-Yates & Ribeiro-Neto, 1999) between the two vectors. A score of zero would indicate that all FVs are unique, which is hardly possible since the approach requires similarity among the entities to be able to populate an ontology. In general, we would like this score to be as low as possible in order to discriminate the FVs,

but this depends a lot on the quality of the ontology.

The next similarity score is the Average Fv Neighbourhood Similarity (AFvNS) that indicates the degree of overlap (semantic relatedness) between neighbouring entities. AFvNS is defined as follows:

Definition 5 *Average Fv Neighbourhood Similarity (AFvNS)*

$$AFvNS(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m sim(fv_i, fv_j)$$

where n is the number of fv s in the ontology o and m is the number of neighbouring entities with fv s of entity i with fv_i . The range is $[0, \dots, 1]$. Note that $AFvS \leq AFvNS$, and, as for AFvS, AFvNS is highly dependent upon the ontology quality.

To evaluate the quality of the FVs, the Normalised Google Distance (NGD) (Cilibrasi & Vitanyi, 2007) is used to compute the semantic distance between an entity and its FV terms. The NGD equation (Cilibrasi & Vitanyi, 2007) is provided below for clarity:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

where $f(x)$ denotes the number of pages containing x and $f(y)$ for y , and $f(x, y)$ denotes the number of pages containing both x and y . N denotes the ‘total number’ of pages in the assumed index by Google™ (set to 20 billion since at this magnitude the precise number of pages is not significant). The range of NGD is between zero and ∞ , where zero denotes best match. However, in practice, most values are in the range of $0, \dots, 1$. Consequently, for the special case where $NGD(x, y) > 1$ we set $NGD(x, y) = 1$. The motivation behind this is that the distance is too large to be of any interest anyway. Note that for this assumption to be valid the constant N must be set to a representative value.

NGD is used in the Average Fv NGD (AFvNGD) score that indicates the semantic distance between an entity and its FV terms. Note, NGD is symmetric by definition, but searches with Google™ are not (e.g., a search for "x y" often yields different results from "y x"). This is tackled by ordering the search terms (for instance, putting the parent entity before a child entity). AFvNGD is defined as follows:

Definition 6 *Average Fv NGD (AFvNGD)*

$$AFvNGD(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m NGD(fv_i, kp_j)$$

where n is the number of fv s in the ontology o and m is the length of the fv_i and fv_i is the name of the entity, and kp_j are the key-phrases of fv_i . Note, if an entity has a parent, then the name of the parent is also included to provide a more specific similarity distance (adapted from [Bouquet et al., 2003] that in our case is limited to the closest parent). $FvNGD(fv)$ will have a score in the same range as NGD.

Once we have found the AFvS, AFvNS and the AFvNGD scores for an ontology, the total score can be calculated. The total Fv Quality Score (FvQS) is an aggregated score of three measures. FvQS provides the overall quality of the FVs and is defined as follows:

Definition 7 *Fv Quality Score (FvQS)*

$FvQS(o) = \alpha(1 - AFvS) + \beta AFvNS + \gamma(1 - AFvNGD)$ where $\alpha + \beta + \gamma = 1$ are weight factors (defaults are 1/3). The total Fv quality score for an ontology will be in the range 0-1, where 1 indicates the best score.

3.4 Restrictions

The evaluation has restrictions as follows:

- All OWL object properties are treated as *other* relations.
- Disjointed classes as a feature are ignored since we do not consider siblings in this evaluation.
- The following equality features are ignored: `equivalentClass`, `sameAs`, and `differentFrom`.
- The maximum length of the FVs has been set to 30 (top 30 selected by highest frequency). In earlier experiments, with no restrictions on FV length, the average length was 24.
- Google™ has a limitation of 64 search results when using the Google™ AJAX Search API. Consequently, we could not use Google™ to test the effect of using more than 64 search results.
- For query expansion, there was a limitation of a maximum of three entities (selected randomly) from each of the possible neighbouring relation types (*parents*, *children*,

and *others*), implying query expansion by a maximum of nine entities in total.

4 Results and analysis

In this section, the results of the experiments are presented and analysed.

4.1 Results

Table 3 summarises the test results of the experiments. In total, 32 experiments were

Table 3 Experimental results.

	AFvS			AFvNS			AFvNGDS			FvQS		
	Animals	Travel	Wine	Animals	Travel	Wine	Animals	Travel	Wine	Animals	Travel	Wine
Bl#1	0,019	0,019	0,040	0,154	0,186	0,286	0,266	0,253	0,163	0,623	0,638	0,694
Bl#2	0,020	0,023	0,041	0,168	0,147	0,286	0,255	0,253	0,180	0,631	0,624	0,688
Ex#1a	0,189	0,138	0,245	0,216	0,178	0,352	0,263	0,233	0,173	0,588	0,602	0,645
Ex#1b	0,107	0,108	0,232	0,175	0,145	0,341	0,257	0,241	0,177	0,604	0,598	0,644
Ex#2	0,048	0,042	0,079	0,304	0,326	0,412	0,194	0,227	0,149	0,687	0,686	0,728
Ex#3	0,021	0,021	0,046	0,288	0,313	0,322	0,277	0,254	0,155	0,663	0,679	0,707
Ex#4	0,021	0,020	0,041	0,178	0,139	0,304	0,265	0,241	0,152	0,631	0,626	0,704
Ex#5	0,040	0,035	0,075	0,404	0,243	0,403	0,214	0,231	0,150	0,717	0,659	0,726
Ex#6	0,048	0,041	0,079	0,288	0,334	0,409	0,200	0,231	0,149	0,680	0,687	0,727
Ex#7	0,020	0,021	0,043	0,278	0,259	0,316	0,276	0,258	0,158	0,661	0,660	0,705
Ex#8	0,039	0,034	0,073	0,406	0,272	0,412	0,215	0,233	0,149	0,717	0,668	0,730
Ex#9	0,015	0,049	0,102	0,211	0,239	0,458	0,261	0,246	0,192	0,645	0,648	0,722
Ex#10	0,019	0,019	0,042	0,130	0,092	0,277	0,270	0,241	0,177	0,613	0,611	0,686
Ex#11	0,014	0,049	0,099	0,224	0,225	0,446	0,249	0,229	0,196	0,654	0,649	0,717
Ex#12	0,031	0,028	0,049	0,187	0,141	0,374	0,268	0,243	0,178	0,629	0,624	0,716
Ex#13	0,015	0,019	0,039	0,127	0,118	0,255	0,250	0,240	0,182	0,621	0,619	0,678
Ex#14	0,017	0,022	0,045	0,210	0,201	0,386	0,259	0,241	0,184	0,644	0,646	0,719
Ex#15	0,015	0,026	0,054	0,221	0,233	0,397	0,268	0,249	0,185	0,646	0,652	0,720
Ex#16	0,022	0,029	0,079	0,070	0,177	0,345	0,149	0,249	0,208	0,633	0,633	0,686
Ex#17		0,014	0,048		0,195	0,293		0,192	0,268		0,663	0,659
Ex#18	0,018	0,026	0,045	0,181	0,136	0,308	0,260	0,236	0,156	0,635	0,625	0,703
Ex#19	0,019	0,025	0,043	0,180	0,132	0,307	0,259	0,226	0,152	0,634	0,627	0,704
Ex#20	0,010	0,023	0,059	0,030	0,151	0,337	0,150	0,258	0,241	0,623	0,623	0,679
Ex#21	0,018	0,023	0,042	0,172	0,137	0,311	0,231	0,234	0,180	0,641	0,627	0,696
Ex#22	0,019	0,027	0,061	0,164	0,112	0,245	0,237	0,220	0,170	0,636	0,621	0,672
Ex#23	0,020	0,022	0,042	0,153	0,093	0,284	0,251	0,229	0,208	0,627	0,614	0,678
Ex#24	0,021	0,025	0,064	0,173	0,069	0,240	0,276	0,251	0,168	0,625	0,598	0,669
Ex#25	0,021	0,019	0,042	0,220	0,153	0,271	0,264	0,257	0,179	0,645	0,626	0,684
Ex#26	0,044	0,043	0,101	0,487	0,343	0,553	0,198	0,237	0,174	0,749	0,687	0,759
Ex#27	0,043	0,046	0,134	0,495	0,320	0,442	0,209	0,225	0,169	0,748	0,683	0,713
Ex#28	0,182	0,161	0,286	0,280	0,253	0,452	0,262	0,243	0,182	0,612	0,616	0,661
Ex#29	0,133	0,098	0,218	0,358	0,261	0,467	0,197	0,220	0,170	0,676	0,648	0,693

Table 5 depicts an example of a FV for the 'bunjee jumping' entity (note that 'bunjee' is written erroneously in the ontology) as part of the Travel ontology. We can observe that 'world' is the second term in Table 5. One can wonder why 'world' is

conducted. The experiments were performed on a standard PC with an Intel™ Pentium processor running Windows™ XP, running Apache Tomcat. Populating the ontologies took more than 29 hours; the most complex ontology, the Wine ontology, took from 10 to 323 minutes to populate. When evaluating the quality of the FVs using NGD, more than 670.000 queries were submitted.

associated with 'bunjee jumping'? By using Yahoo!™ we find that, 'bunjee jumping' and 'world' coexist in more than 13.000 documents (e.g., 'world's highest', 'world's largest'). Consequently, they are highly related on the Web.

Table 4 FvQS relative to the baseline.

	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10	Ex#11	Ex#12	Ex#13	Ex#14	Ex#15
Animals ontology	9,1%	5,3%	0,0%	13,8%	7,9%	4,8%	13,9%	2,3%	-2,8%	3,7%	-0,2%	-1,6%	2,2%	2,4%
Travel ontology	9,7%	8,7%	0,4%	5,5%	10,0%	5,7%	7,0%	3,8%	-2,0%	4,0%	0,0%	-0,6%	3,5%	4,5%
Wine ontology	5,7%	2,7%	2,2%	5,5%	5,6%	2,4%	6,0%	4,8%	-0,3%	4,1%	4,0%	-1,4%	4,4%	4,5%
Average	8,2%	5,6%	0,9%	8,3%	7,8%	4,3%	9,0%	3,6%	-1,7%	4,0%	1,3%	-1,2%	3,4%	3,8%

	Ex#16	Ex#17	Ex#18	Ex#19	Ex#20	Ex#21	Ex#22	Ex#23	Ex#24	Ex#25	Ex#26	Ex#27	Ex#28	Ex#29
Animals ontology	0,4%		0,6%	0,6%	-1,2%	1,7%	0,8%	-0,5%	-0,9%	2,3%	18,9%	18,8%	-3,0%	7,3%
Travel ontology	1,5%	6,2%	0,2%	0,5%	0,0%	0,5%	-0,3%	-1,5%	-4,1%	0,3%	10,0%	9,3%	-1,1%	3,8%
Wine ontology	-0,3%	-4,2%	2,1%	2,2%	-1,3%	1,1%	-2,4%	-1,5%	-2,7%	-0,7%	10,2%	3,6%	-3,9%	0,7%
Average	0,5%	1,0%	1,0%	1,1%	-0,9%	1,1%	-0,6%	-1,2%	-2,6%	0,7%	13,1%	10,6%	-2,7%	3,9%

Figure 2 A graphical representation of the FvQS relative to the baseline.

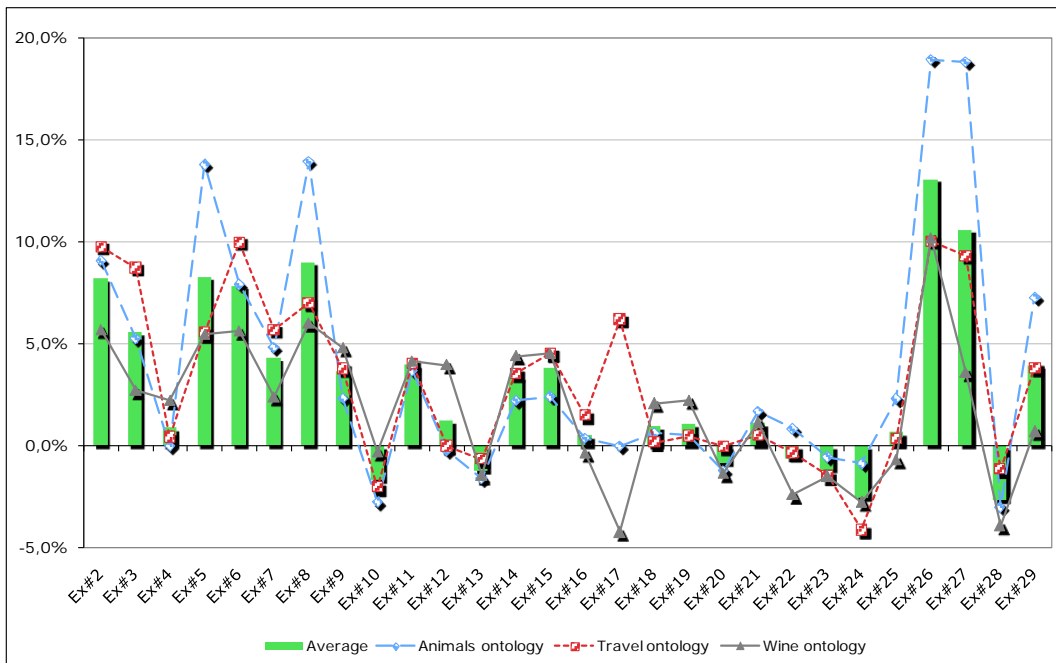


Table 5 A 'bunjee jumping' FV example from the Travel ontology.

Key-phrase	Freq.	Key-phrase	Freq.
bunjee	0,226	clubs	0,019
world	0,075	fun	0,019
bunjee jump	0,057	giant	0,019
adventures	0,038	informaiton bunjee	0,019
bungee	0,038	lists	0,019
bungee jump	0,038	peak	0,019
bunjee cliff	0,038	place	0,019
jump	0,038	rebel	0,019
nepal bunjee	0,038	rebel billionaire	0,019
world cup	0,038	resource	0,019
activities world	0,019	south	0,019
adventure activities	0,019	south africa	0,019
backyard bunjees	0,019	video	0,019
billionaire giant	0,019	video sites	0,019
clips	0,019	world heritage	0,019

4.2 Analysis

An overview of the experiments and their percentage difference relative to the baseline is

shown in Table 4. Figure 2 provides a graphical representation of the results in Table 4. Next, we provide some comments about the findings of the experiments:

- *Ontology analysis (Ex#22, 24):* Ex#22 scored on average slightly worse than the baseline. A surprise is the increase in score for the Animals ontology, since with or without the use of a reasoner, the results for this ontology should be the same. The reason for this increase is probably the same as indicated for the Ex#1a&b experiments. Another unexpected result was the decrease in score for the Travel and Wine ontologies. A reason for this might be that the additional relations provided by reasoning are not fully utilised, which we test in Ex#24 and Ex#27. However, if we look at the error rates (the ratio of entities populated versus not

populated) in Table 6 we see a decrease in errors for the Travel and Wine ontologies, which can indicate that more relations provided by the reasoner helps to populate more entities.

- *Ranking of entities (Ex#23, 24):* Another surprise was the slight decrease in the FV quality score in both Ex#23 and Ex#24 when ranking the entities. This is probably because no entities (other than the parent entities) were used when identifying the most prominent cluster candidates. However, in Ex#27 children entities were also used but still we have the same tendency, a lower score than the comparable test Ex#26. If we look at the error rates in Table 6, we see an increase in errors as well. These observations combined can indicate the drifting problem described in Step 2 (Section 2).
- *Query expansion - neighbours (Ex#2-8):* Ex#8 provided on average the best results, and the best results for the Animals and the Wine ontologies. However, for the Travel ontology Ex#8 provided the fourth best results while Ex#6 gave the best results for this ontology. It was assumed that Ex#2 on average would provide the best results but it turned out that it provided the third best results. If we look at both the standard deviation and mean results, then Ex#2 yields the best results. This could indicate that independent of the quality of the ontology, Ex#2 would be the best choice.
- *Query expansion - keywords (Ex#28, 29):* The results from Ex#28 indicate that adding global keywords is not beneficial w.r.t. the overall FV quality score. The AFvS score is high for both Ex#28 and Ex#29, which is also the case for Ex#1a&b. Ex#29 indicates an increase but compared to Ex#2 it is a decrease. However, as discussed in Section 2 Step 3, homogeneous FVs can be a feature that is beneficial depending on the intended usage.
- *Search engine (Ex#13, 25):* As can be seen from the results, changing the search engine does influence the results, in this case negatively. This was a bit surprising since the algorithm is not tailored to any particular search engine. By experience, the search results are on average equal for the two search engines used in this experiment, mainly the ranking of the documents is what differs. This explains the positive effect of increasing the search results in Ex#25. Earlier experiments have shown that the snippet lengths of the two search engines are equal in length as well. However, we have not checked the difference in quality of the snippets, which might be the cause.
- *Number of search results (Ex#14, 15):* In Ex#14 and Ex#15 we tested if the number of search results retrieved and processed would affect the FV quality, which is the case with an increase of 3,4% and 3,8% respectively. More clusters are more expensive to compute. In Ex#2, the Animals, Travel and Wine ontologies took 3, 3, and 16 minutes to process respectively while Ex#14 took on average 3 times as long to process and Ex#15 took 7 times as long. In this experiment, we have not tried to find the optimal number of results to process, but just by looking at the increase of the FV quality from 30 to 100 results versus 200 results indicates that 100 is the optimal number with regards to both the FV quality and the processing time.
- *Content (Ex#9):* The results of Ex#9 show a slight improvement with an average of 3,6% compared to the baseline. It is uncertain if this result is optimal since we have experienced some difficulties using full text documents. Many sites do not allow the direct download of Web pages for other purposes than browsing. Consequently, some of the documents became unavailable; this might influence the quality of the FVs. Nevertheless, Ex#9 showed an improvement compared to the baseline.
- *Clustering - input (Ex#10, 11):* In Ex#11 we tested if it is more beneficial to use document FVs (DFV) as input to the clustering algorithms or snippets. Ex#11 showed some improvement of using DFVs compared to Ex#9 with only 0,4%, probably because the DFVs are more focused by extracting only those parts of the documents considered most relevant to the search. However, when creating DFVs for the snippets, Ex#10 showed a decrease in performance by 1,7% indicating that the snippets are best used as is.
- *Clustering - algorithms (Ex#12):* In Ex#12 we tested to see if changing to the

STC clustering algorithm would influence the FV quality, which it did with our clustering algorithm settings. STC got an increase of 1,3% compared to Lingo. Note we have not fine-tuned the settings of either Lingo or STC. Consequently, we cannot conclude at this stage whether STC is better than Lingo.

- *Domain identification (Ex#16-21):* Not surprisingly, we got more or less the same results as for the query expansion experiments (Ex#2-Ex#8) where using *parents*, *children*, and *other* neighbouring entities provided the best results (Ex#21). Ex#19 got the same results as Ex#21 but with a higher standard deviation indicating that Ex#21 provides better results independent of the ontology quality. Ex#18 and Ex#19 provide more or less the same results. For Ex#16, Ex#17, and Ex#20 the algorithm failed to populate most of the entities (see Table 6 and Figure 3). In fact, for Ex#17 no entities were populated for the Animals ontology since the ontology only got super- and sub-class relationships and hence no *other* relations. Consequently, the results from Ex#16, Ex#17, and Ex#20 can be disregarded.

- *Best practice (Ex#26, 27):* These experiments were conducted to test the combination of some of the best results from the other experiments. Both Ex#26 and Ex#27 performed considerably better than the other experiments with an increase of 13,1% and 10,6% respectively. The system performed considerably worse for the Wine ontology in Ex#27 than for Ex#26; this is probably due to the ranking problem as described for Ex#22 and Ex#24. However, if we look at the error rates (Table 6) for Ex#27, this is very low indicating that additional relations provided by a reasoner have a positive effect on populating more entities (the same findings as for Ex#22).

In addition, we tested what effect a time span of one year had on the quality of the FVs (Ex#1a vs. b). The quality of the FVs for Ex#1b had an increase of 2,7% for the Animals ontology, while a slight decrease for the Travel and Wine ontology when we compared it with the results from Ex#1a. We observe the same trend for the baseline. However, the difference was less than expected when compared to the baseline.

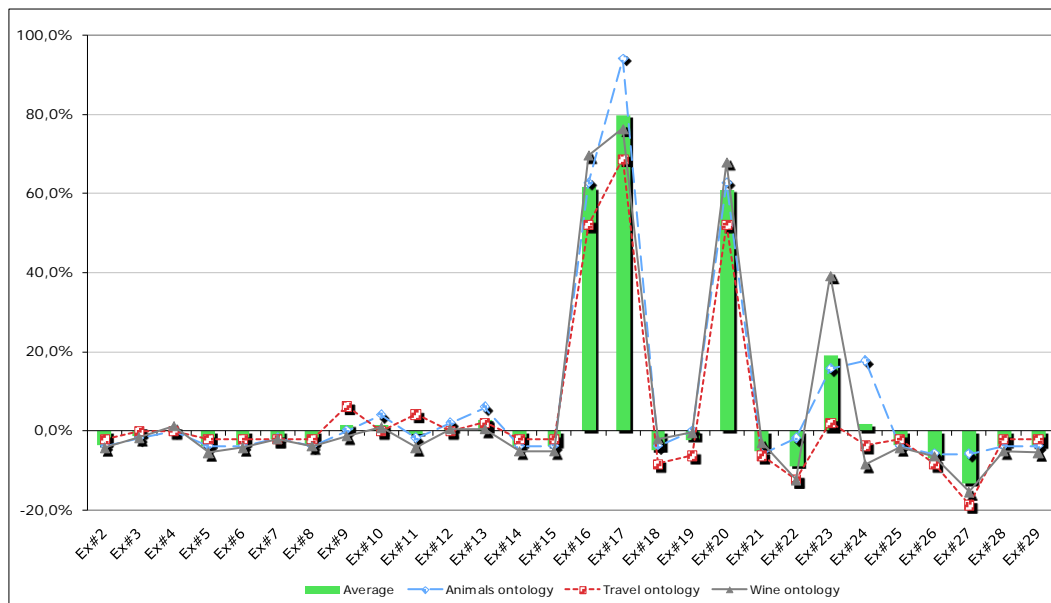
Table 6 Error rate analysis (ratio of entities populated versus not populated).

	Ex#2	Ex#3	Ex#4	Ex#5	Ex#6	Ex#7	Ex#8	Ex#9	Ex#10	Ex#11	Ex#12	Ex#13	Ex#14	Ex#15
Animals ontology	-3,9%	-2,0%	0,0%	-3,9%	-3,9%	-2,0%	-3,9%	0,0%	3,9%	-2,0%	2,0%	5,9%	-3,9%	-3,9%
Travel ontology	-2,1%	0,0%	0,0%	-2,1%	-2,1%	-2,1%	-2,1%	6,3%	0,0%	4,2%	0,0%	2,1%	-2,1%	-2,1%
Wine ontology	-4,2%	-1,7%	1,3%	-5,5%	-4,2%	-2,1%	-3,8%	-1,3%	0,8%	-4,2%	0,4%	0,4%	-5,1%	-5,1%
Average	-3,4%	-1,2%	0,4%	-3,8%	-3,4%	-2,1%	-3,3%	1,7%	1,6%	-0,7%	0,8%	2,8%	-3,7%	-3,7%
	Ex#16	Ex#17	Ex#18	Ex#19	Ex#20	Ex#21	Ex#22	Ex#23	Ex#24	Ex#25	Ex#26	Ex#27	Ex#28	Ex#29
Animals ontology	62,7%	94,1%	-3,9%	0,0%	62,7%	-5,9%	-2,0%	15,7%	17,6%	-3,9%	-5,9%	-5,9%	-3,9%	-3,9%
Travel ontology	52,1%	68,8%	-8,3%	-6,3%	52,1%	-6,3%	-12,2%	2,1%	-3,7%	-2,1%	-8,3%	-18,6%	-2,1%	-2,1%
Wine ontology	69,6%	76,4%	-2,1%	-0,4%	67,9%	-3,0%	-12,3%	39,2%	-8,4%	-4,2%	-6,3%	-15,4%	-5,1%	-5,5%
Average	61,5%	79,7%	-4,8%	-2,2%	60,9%	-5,0%	-8,8%	19,0%	1,8%	-3,4%	-6,8%	-13,3%	-3,7%	-3,8%

Table 6 and Figure 3 present the error rates. The error rate is the ratio of entities populated versus not populated. For most of the experiments, a

slight decrease in errors is observed. However, for Ex#16, 17, 20, and 23 the errors increased as previously explained for these experiments.

Figure 3 Population error rate relative to the baseline.



4.3 Key findings

Based on the findings in the conducted experiments we conclude the following:

- *Step 1*: Using a reasoner lowers the error rate, but can decrease the overall FV quality.
- *Step 2*: Ranking of entities for processing seems to decrease the FV quality and increase the error rate. Consequently, doing this is not a recommended practice.
- *Step 3*: Query expansion increases the quality of the search results and hence the FV quality. Including the parents, children, and other related entities provide the best results.
- *Step 4*: Change of comparable search engines does not seem to yield any major effect if an adequate number of search results is used.
- *Step 5 and 6*: Using full text documents in combination with the extraction of the most relevant key-phrases seems to provide the best positive effect on the FV quality. However, this increases the processing time considerably compared to using just snippets (probably due to the downloading of each page).
- *Step 6*: Comparable clustering algorithms do not seem to yield any major effect.

- *Step 7*: Including the parents, children, and other related entities seems to provide the best results when identifying the most prominent cluster candidates.

The most important component w.r.t. the FV quality is the query expansion component (Step 3). The parent entities are the most important neighbouring entities both for query expansion (Step 3) and when identifying the most prominent candidate cluster (Step 7). Further, the neighbouring entities used to expand the query yields better FV quality than usage of scope keywords. A high number of search results minimises the difference between the search engines and probably the change in ranking they provide over time.

Further, we have some interesting observations from Table 4. The Animals ontology (taxonomy) is most sensitive to different techniques, i.e. for this type of ontology, a certain combination of techniques may radically improve results (e.g., Ex#5, Ex#8, and Ex#26 have the biggest improvements w.r.t. other ontologies).

5 Related work

In this section, we explore related work on the construction of feature vectors (FV). FVs can in general be classified in three groups, numerical,

textual, and a mix of each. Numerical FVs are typically used in machine learning (e.g. Scuturici et al., 2005; Mitchell, 1997), and will not be included in this overview. We will not include approaches using mixed FVs. Textual FVs on the other hand, are typically based on a lexical resource like WordNet (e.g. Lopez et al., 2006) or extracted from a set of documents (e.g. Agirre et al., 2000; Su & Gulla, 2006; Gabrilovich & Markovitch, 2007; Solskinnsbakk & Gulla, 2008). There are also approaches that assume FVs already are created (e.g. Formica et al., 2008) and consequently focus on the usage of FVs; these approaches will not be considered in this overview. Next, a set of approaches related to our work is analysed.

There are approaches computing the semantic relatedness of concepts with similarities to our approach (e.g. Gabrilovich & Markovitch, 2007; Kulkarni & Caragea, 2009). Gabrilovich and Markovitch (2007) utilise the vast amount of organized human knowledge that is available in knowledge repositories like Wikipedia and Open Directory Project (ODP). Each node in ODP is treated as a concept. A textual object is created for each node consisting of concatenated Web documents (listed for each node by ODP) and their textual descriptions. The concepts are represented as attribute vectors. A document is divided into non-overlapping segments called contexts where each context is classified into one or several concepts. An ambiguous concept will be part of several domains, which is partly resolved by categorising them. In the case of hierarchies, a parent node will typically consist of both the child concepts and a textual description.

Kulkarni and Caragea (2009) propose a Concept Extractor and Relationship Identifier (CE-RI) system to bridge the gap between the current Web and the Semantic Web. The Concept Extractor (CE) component is relevant to our work. As Gabrilovich and Markovitch (2007) CE exploits the vast amount of information found on the Web but in contrast does not rely on a knowledge base like Wikipedia. They utilise the power of existing search engines to collect a set of documents relevant to a set of queries based on the user query. Then they use PageRank (Page et al., 1999) in combination with the document frequencies to find the most representative documents w.r.t. the user query. Based on these documents, they extract a set of concepts. However, instead of extracting a set of terms from the documents (in contrast to our approach) they

rely on meta information being available, more specifically, meta keywords and the titles of the Web pages. It is unclear how vulnerable this approach is with respect to ambiguous words.

Approaches based on topic signatures are similar in spirit to our approach. A topic signature is a list of topically related words (Agirre et al., 2000). There are many topic signature approaches (e.g., Agirre et al., 2000; Zhou et al., 2007). Zhou et al. (2007) propose a Topic Signature Language Model that is used to perform semantic smoothing to increase retrieval performance. They create topic signatures for each concept defined in a domain specific ontology using a highly relevant document collection. The topic signature terms are found by collocation. They assume the concepts are unique and consequently circumvent the problem of word disambiguation. For general domains where no ontology exists, they propose to use multiword expressions as topic signatures. The multiword expressions contain context and are consequently mostly unambiguous.

Agirre et al. (2000) propose enriching WordNet with topic signatures using the Web. A concept in WordNet can contain several senses. Nevertheless, for each sense a set of cue-words (hyponyms, hypernyms, etc.) is used to create a highly specific query that is submitted to the search engine. The top 100 documents are retrieved and keywords are extracted. They experienced formulating the queries as being the weakest point of their approach. The quality of the queries highly affected the quality of the retrieved documents. This is in contrast to our approach that is not dependent upon a high quality query but uses clustering and domain identification by utilising neighbouring entities to find relevant documents from a set of diverse documents.

Unfortunately, evaluation of the quality of feature vectors and topic signatures is scarce. Mostly they are evaluated indirectly based on performance of a designated application. Therefore, we hope that the method reported here will inspire others to endeavour more detailed evaluation of their approaches.

6 Conclusions and future work

In this study, we have evaluated the sensitivity of the components of a feature vector construction approach. The overall construction process has been briefly described analysing its components w.r.t. both the intrinsic FV quality and three ontologies used. In total, 32 experiments were

conducted. Based on the evaluation of these experiments we have concluded what components contribute most positively (the query expansion and domain identification components) to the FV quality. The contribution of this paper is a presentation of an evaluation method and lessons learnt. We have shown that some choices, when implementing components, impact the quality of the resulting FVs and, finally, the performance of the systems.

We have not been able to test the optimised feature vectors, based on the findings in this experiment, in our search application. Nevertheless, based on the type of evaluation, e.g. using the Web to evaluate the FV quality, and earlier experiments, we can extrapolate (with high confidence) an increase in the retrieval effectiveness of applying these findings to our search system. However, this needs to be empirically confirmed.

Limited number of used ontologies does not allow generalising the results. Therefore, one of the future tasks is to conduct a similar experiment with more and bigger ontologies. We need to categorise the ontologies according to different key characteristics to find trends relevant to the categories. We have done some minor experiments with the NGD measure to assess the semantic distance among the entities of the ontologies used in this experiment. Preliminary results indicate that there is a connection between the individual findings of the ontologies in this experiment and the NGD ontology score. This needs to be explored further.

Acknowledgments

This research work is partially funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30. In addition, we would like to thank Jon Atle Gulla (NTNU), Per Gunnar Auran (Yahoo!), and Robert Engels (ESIS) for their support and help. Further, we would like to thank the reviewers for their helpful comments and suggestions.

References

- Agirre, E., Ansa, O., Hovy, E.H. & Martinez, D. (2000) 'Enriching very large ontologies using the WWW'. *In: Proceedings of ECAI Workshop on Ontology Learning*.
- Alani, H., Brewster, C. & Shadbolt, N. (2006) 'Ranking Ontologies with AKTiveRank'. *In: The Semantic Web - ISWC 2006*, LNCS 4273, (pp.1-15), Heidelberg: Springer-Verlag.
- Baeza-Yates, R. & Ribeiro-Neto, B. (1999) *Modern information retrieval*, New York: ACM Press.
- Berners-Lee, T., Hendler, J. & Lassila, O. (2001) 'The Semantic Web'. *Scientific American*, Vol. 284, No. 5, pp. 34-43.
- Bouquet, P., Serafini, L. & Zanobini, S. (2003) 'Semantic Coordination: A New Approach and an Application'. *In: The SemanticWeb - ISWC 2003*, LNCS 2870, (pp.130-145), Heidelberg: Springer-Verlag.
- Carrot2 (2009). Carrot2 an Open Source Search Results Clustering Engine. Obtained through the Internet: <http://www.carrot2.org/>, [accessed 27/05/2009].
- Castells, P., Fernandez, M., & Vallet, D. (2007) An adaptation of the vector-space model for ontology-based information retrieval. *IEEE TKDE*, Vol. 19, No. 2, pp.261-272.
- Cilibrasi, R. & Vitanyi, P. (2007) 'The Google Similarity Distance'. *IEEE TKDE*, Vol. 19, No. 3, pp.370-383.
- d'Aquin, M. et al. (2008) 'Toward a New Generation of Semantic Web Applications'. *IEEE Intelligent Systems*, Vol. 23, No. 3, pp.20-28.
- Formica, A., Missikoff, M., Pourabbas, E. & Taglino, F. (2008) 'Weighted Ontology for Semantic Search'. *In: R Meersman & Z Tari (eds), On the Move to Meaningful Internet Systems: OTM 2008*, LNCS 5332, (pp.1289-1303), Heidelberg: Springer-Verlag.
- Gabrilovich, E. & Markovitch, S. (2007) 'Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization'. *J. Mach. Learn. Res.*, Vol. 8, pp.2297-2345.
- Gligorov, R., Aleksovski, Z., ten Kate, W. & van Harmelen, F. (2007) 'Using Google distance to weight approximate ontology matches', *In: WWW '07: Proc. of the 16th int. conf. on WWW*, ACM Press.
- Jiang, X. & Tan, A.H. (2006) 'OntoSearch: A Full-Text Search Engine for the Semantic Web'. *In: Proc. of the 21st National Conf. on Artificial Intelligence*, (pp.1325-1330). Boston: AAAI Press
- Kulkarni, S. & Caragea, D. (2009) 'Towards Bridging the Web and the Semantic Web'. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2009. WI-IAT '09*, (pp. 667-674), Milano: IEEE Computer Society.
- Lopez, V., Sabou, M. & Motta, E. (2006) 'PowerMap: Mapping the Real Semantic Web on the Fly'. *In: The Semantic Web - ISWC 2006*, LNCS 4273, (pp. 414-427), Heidelberg: Springer-Verlag.
- Mitchell, T. M. (1997) *Machine Learning*, New York, McGraw-Hill.
- Nagyppál, G. (2005) 'Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies'. *In: Meersman, R., Tari, Z. &*

- Herrero, P. (eds.) *On the Move to Meaningful Internet Systems 2005: O TM Workshops*, LNCS 3762, (pp.780-789), Heidelberg: Springer-Verlag.
- Ogden, C.K. & Richards, I.A. (1930) *The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism*, London: Kegan Paul, Trench, Trubner & Co.
- Page, L., Brin, S., Motwani, R. & Winograd, T. (1999) 'The PageRank Citation Ranking: Bringing Order to the Web'. Stanford InfoLab.
- Scuturici, M., Clech, J., Scuturici, V.M. & Zighed D.A. (2005) 'Topological representation model for image database query'. *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 17, No. 1-2, pp.145-160.
- Sirin, E. et al. (2007) 'Pellet: A practical OWL-DL reasoner'. In: *Web Semantics: Science, Services and Agents on the World Wide Web*, (pp.51-53), Amsterdam: Elsevier Science Publishers.
- Solskinnsbakk, G. & Gulla, J. (2008) 'Ontological Profiles as Semantic Domain Representations'. In: *Natural Language and Information Systems*, LNCS 5039, (pp.67-78), Heidelberg: Springer-Verlag.
- Strasunskas, D. & Tomassen, S. L. (2008) 'The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation'. *Int. J. Knowledge and Learning*, Vol. 4, No. 4, pp. 398-414.
- Su, X. & Gulla, J. A. (2006) 'An information retrieval approach to ontology mapping'. *Data & Knowledge Engineering*, Vol. 58, No. 1, pp.47-69.
- Tomassen, S. L. & Strasunskas, D. (2009) 'Construction of Ontology based Semantic-Linguistic Feature Vectors for Searching: the Process and Effect'. In: *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, (pp.133-138), Washington: IEEE Computer Society.
- van Harmelen, F. (2006) 'Semantic Web Research Anno 2006: Main Streams, Popular Fallacies, Current Status and Future Challenges', In: M Kl usch, M Rovatsos & T Payne (eds), *Cooperative Information Agents X*, LNCS 4149, (pp.1-7), Heidelberg: Springer-Verlag.
- W3C (2004). Web Ontology Language (OWL). Obtained through the Internet: <http://www.w3.org/2004/OWL/>, [accessed 27/05/2009].
- Zhou, X., Hu, X. & Zhang, X. (2007) 'Topic Signature Language Models for Ad hoc Retrieval'. *IEEE TKDE*, Vol. 19, No. 9, pp.1276-1287.

P5: Constructing Feature Vectors for search: investigating intrinsic quality impact on search performance

Publication details

Tomassen, S.L. & Strasunskas, D. (2010) Constructing Feature Vectors for search: investigating intrinsic quality impact on search performance. *Int. J. Web and Grid Services*, 6(3), pp. 289-312.

Constructing Feature Vectors for search: investigating intrinsic quality impact on search performance

Stein L. Tomassen*

Dept. of Computer and Information Science
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
E-mail: steint@idi.ntnu.no
* Corresponding author

Darijus Strasunskas

Dept. of Industrial Economics and Technology Management
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
E-mail: darijuss@gmail.com

Abstract: In this paper, we revisit our approach to construction of semantic-linguistic Feature Vectors (FVs) used in search. These FVs are built based on domain semantics encoded in an ontology and enhanced by relevant terminology from Web documents. The contributions of this paper are the evaluation of constructed FVs and the analysis of their impact on search performance. This completes the validation of the proposed approach concluding that the proposed metrics provide good indications of the quality of the FVs. Yet, the results suggest the metrics need to be revised to fit the needs of search applications.

Keywords: ontology; FVC; feature vector construction; evaluation; validation; search performance; intrinsic quality.

1 Introduction

The Web is becoming a dominant information repository. However, retrieval of relevant information is still a challenging task for most of its users. Ambiguity of words is one of the main hindrances in information retrieval (e.g., Bhogal et al., 2007; Carmel et al., 2006). Employment of semantic technologies in search systems is seen as a promising approach to improve the current state of the art (e.g., Horrocks, 2007). Semantic technologies are applied in different ways: semantic annotations of content (e.g., Moscato et al., 2009); clustering of retrieved documents according to topics (e.g., Panagis et al., 2006); powerful querying languages (e.g., Bry et al., 2005); or creating structured semantic models of retrieved documents (e.g., Noah et al., 2005). In summary, many efforts are devoted to improve information retrieval (IR) using ontologies, for instance, (Bhogal et al., 2007; Castells et al., 2007; Suomela & Kekalainen, 2005).

The objective of this paper is to validate a developed approach to semantic search that builds on a concept of feature vectors (FV). The approach is based on a pragmatic use of ontologies by relating the concepts (domain semantics) with the actual terminology used in a text corpus, i.e., the Web. Therefore, we propose to associate every entity (classes and individuals) of the ontologies with a FV to tailor them to the domain terminology in a text corpus. First, these FVs are created off-line and later used on-line to filter, and hence disambiguate search, and re-rank the search results from the underlying search system (Tomassen & Strasunskas, 2009b).

There are three typical objectives for evaluation: a) to prove advantage in performance over existing traditional or competitive approaches; b) understand performance sensitivity of a system by evaluating different configurations of the system; or c) assess usability and user experience of a system. Standard relevance metrics are used to fulfil the first objective when evaluating search systems. However, specificity of semantic search systems requires tailored benchmark datasets, i.e., a set of annotated

documents and relevant queries. The second objective can be pursued by developing evaluation frameworks and intrinsic quality metrics. The frameworks and metrics would allow assessing interdependence of sub-components and deriving a “best-in-breed” configuration of a system. Finally, the third objective is an ultimate goal of any system that is made for end-users. Recent progress and results in the semantic search area indicate an improvement compared to traditional IR systems (e.g., Castells et al., 2007; Formica et al., 2008; Jiang & Tan, 2006; Solskinnsbakk & Gulla, 2008). Yet, the results lack indications whether this improvement is optimal (Strasunskas & Tomassen, 2010) since many evaluations are restricted to the first objective. In (Tomassen & Strasunskas, 2010), we proposed an evaluation method based on an analysis of components' sensitivity with regards to quality of resulting FVs, where the proposed metrics were analytically derived from contemporary literature. The intrinsic quality measure provides a mean to estimate the output of different configurations of the algorithm, yet it needs to be related to actual performance of the search application. Therefore, in this paper we go one step further and investigate performance of the overall approach related to different qualities of FVs, i.e., we validate the intrinsic quality measures presented in (Tomassen & Strasunskas, 2010). The experiment was conducted with real end-users.

This paper is organized as follows. In section 2, related work is discussed. In section 3, the feature vector construction process is described. In section 4, we present the conducted experiment and explain the evaluation. Then in section 5, the results are presented and analyzed. Finally, in section 6, we conclude this paper and sketch future work.

2 Related work

In this section, we provide an overview of related work on enhancement of search by semantics and relate our evaluation to current practice in the field. This literature review is limited to approaches that build on a notion similar to our feature vectors (FV).

2.1 Describing a topic

Feature vectors can, in general, be classified into three groups: numerical, textual, and the combination of both. Numerical FVs are typically used in machine learning (e.g., Mitchell, 1997; Scuturici et al., 2005), and will not be included in this overview. Neither will approaches using mixed FVs be included. Textual FVs, on the other hand, are typically based on a lexical resource like WordNet (e.g., Lopez et al., 2006) or extracted from a set of documents (e.g., Agirre et al., 2000; Gabrilovich & Markovitch, 2007; Solskinnsbakk & Gulla, 2008; Su & Gulla, 2006). In addition, there are approaches assuming already created FVs (e.g., Formica et al., 2008) and consequently focus on the usage of FVs; these approaches will neither be considered in this overview. Next, a set of approaches related to our work is analysed.

Approaches based on topic signatures are similar in spirit to our approach. A topic signature is a list of topically related words (Agirre et al., 2000). There are many topic signature approaches (e.g., Agirre et al., 2000; Zhou et al., 2007). Zhou et al. (2007) propose a Topic Signature Language Model that is used to perform semantic smoothing to increase retrieval performance. They create topic signatures for each concept defined in a domain specific ontology using a highly relevant document collection. The topic signature terms are found by collocation. They assume unique concepts and consequently circumvent the problem of word disambiguation. For general domains, where no ontology exists, they propose to use multiword expressions as topic signatures. The multiword expressions contain context and are consequently mostly unambiguous.

Agirre et al. (2000) propose enriching WordNet with topic signatures using the Web. A concept in WordNet can contain several senses. For each sense, a set of cue-words (hyponyms, hypernyms, etc.) is used to create a highly specific query that is submitted to a search engine. The top 100 documents are retrieved and keywords are extracted. They experienced formulating the queries as being the weakest point of their approach since the quality of the queries highly affected the quality of the retrieved documents. This is in contrast to our approach that is not dependent upon a high quality query but uses clustering and domain identification by utilising neighbouring entities to find relevant documents from a set of diverse documents.

There are approaches computing the semantic relatedness of concepts with similarities to our approach (e.g., Gabrilovich & Markovitch, 2007; Kulkarni & Caragea, 2009). Gabrilovich and Markovitch (2007) utilise the vast amount of organized human knowledge that is available in knowledge repositories like Wikipedia and Open Directory Project (ODP). Each node in ODP is treated as a concept. A textual object is created for each node consisting of concatenated Web documents (listed for each node by ODP) and their textual descriptions. The concepts are represented as attribute vectors. A document is divided into non-overlapping segments called contexts where each context is classified into one or several concepts. An ambiguous concept will be part of several domains, which is partly resolved by categorising them. In the case of hierarchies, a parent node will typically consist of both the child concepts and a textual description.

Kulkarni and Caragea (2009) propose a Concept Extractor and Relationship Identifier (CE-RI) system to bridge the gap between the current Web and the Semantic Web. The Concept Extractor (CE) component is relevant to our work. Kulkarni and Caragea (2009) exploits the vast amount of information found on the Web but does not rely on a knowledge base like Wikipedia as Gabrilovich and Markovitch (2007) did. They utilise the power of existing search engines to collect a set of documents relevant to a set of queries based on the user query. Then they use PageRank (Page et al., 1999) in combination with the document frequencies to find the most representative documents for the user query. Based on these documents, they extract a set of concepts. However, instead of extracting a set of terms from the documents, in contrast to our approach, they rely on meta information being available, more specifically, meta keywords and the titles of the Web pages. It is unclear how vulnerable this approach is with respect to ambiguous words.

2.2 *Evaluating semantic search systems*

Evaluation methods in information retrieval are typically classified as system-centric and user-centric. Methods in the former category are based on or derived from precision and recall metrics (Baeza-Yates & Ribeiro-Neto, 1999). However, these metrics are criticized for not being able to indicate the causes for variation of different retrieval results that remain hidden under the average recall and precision figures (Alemayehu, 2003). User-centric evaluations, on the other hand, try to assess the probability of an IR system being adopted and used. When taking a closer look at evaluation of semantic search systems, we notice a lack of end-users' involvement (e.g., Castells et al., 2007; Wang et al., 2008; Zhang et al., 2005).

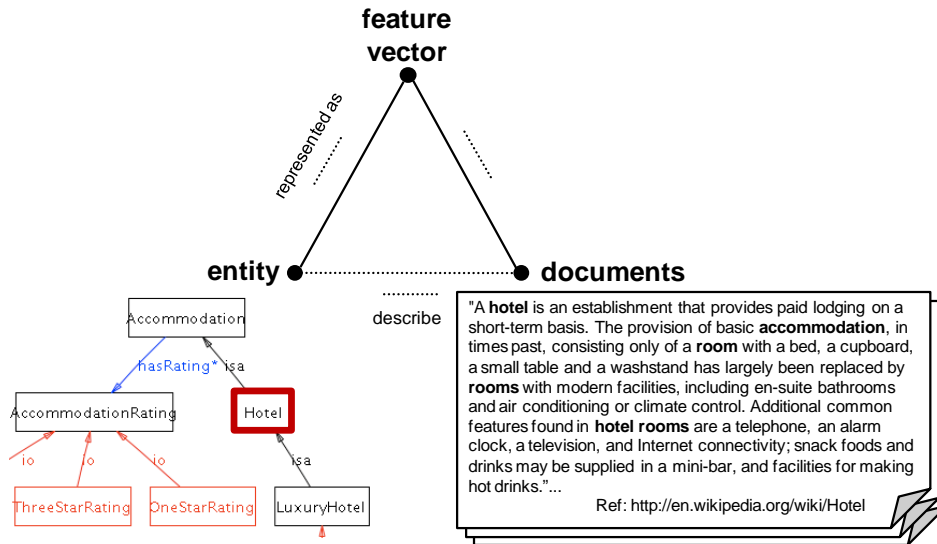
Dominance of “black-box” approaches (where only output of system is measured) and insufficient end-users involvement has motivated us for a thorough evaluation of our semantic search approach. Therefore, here we establish a frame of reference for the evaluation reported in this paper in comparison to our earlier experiments. In (Tomassen & Strasunskas, 2010) we focused on aspects of the FV construction algorithm components and their affect on the FV quality. We focused on the FV construction process since the actual search performance depends a lot on the quality of FVs. In (Strasunskas & Tomassen, 2008), we investigated FVs use in search to disambiguate queries that was evaluated with real users and reported on average an improvement of search by more than 10%. In (Strasunskas & Tomassen, 2008) we found significant dependence between overall performance and ontology quality. However, we were not able to conclude to what degree FV quality depends on ontology and how much it is influenced by the FV construction process and the techniques used there. Therefore, in (Tomassen & Strasunskas, 2010) we focused on the FV construction algorithm and its affect on the FV quality, while the actual performance of the assumed best FV quality parameters remain untested. Next, we briefly introduce the approach before diving into details of the experiment.

3 **Feature vector construction**

Every ontology entity (class or individual) has a feature vector (FV) with a set of associated terms extracted from a text corpus. In this section, we describe the process of how these FVs are constructed. We present an overall overview of the construction process (more details are found in (Tomassen & Strasunskas, 2009a, 2010)), but first we provide an introduction to FVs as follows.

Figure 1 An illustration of the relationship between a feature vector, an entity, and a set of documents.

Hotel = {bonus cash, book, brand-name hotels, cash book, cheap hotels, discount hotel deals, discount hotel rooms, guarantee, hotel, hotel bonus, hotel rooms, hotels motels resorts, hotwire, independent hotel, low price guarantee, other accomodations, popular cities, price guarantee, quality name brand, right accommodation, rooms, same time}



3.1 Introduction to Feature Vectors

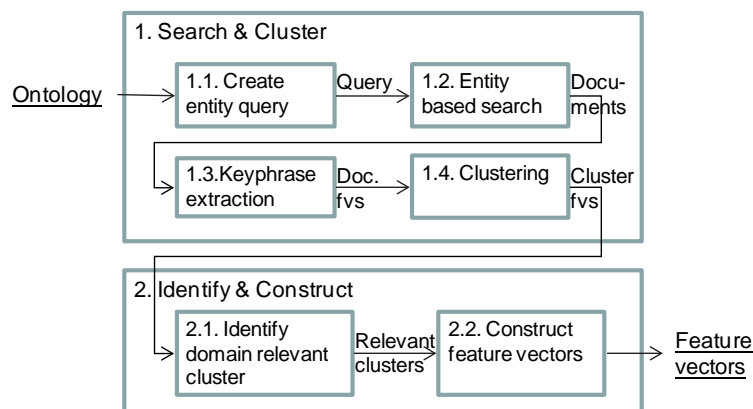
A feature vector "connects" a concept (entity) to a document collection, i.e., a FV is tailored to the specific terminology used in a particular document collection. FVs are built considering both the semantics encoded in an ontology and the dominant lexical terminology surrounding the concepts (entities) in a text corpus. The underlying idea is that a FV reflects both the semantic and linguistic neighbourhoods of a particular entity. The semantic neighbourhood is computed based on related entities and direct properties specified in an ontology (or a fragment of an ontology in case of a broad ontology (Bhatt et al., 2004)), while the linguistic neighbourhood is based on co-location of terms in a document collection. Therefore, a FV constitutes a rich representation of an entity that is related to the actual terminology used in a text corpus. Figure 1 shows an illustration of a FV and how it relates to an entity and a set of documents. For a more formal definition of a FV, the keen reader is referred to (Tomassen & Strasunskas, 2010).

3.2 Construction of Feature Vectors

The Feature Vector Construction (FVC) algorithm is presented in detail in (Tomassen & Strasunskas, 2009a). However, to make this paper self-contained and to provide a basis for the experiments presented in section 4, the algorithm is outlined here as well.

The FVC process is visualized in Figure 2 with an illustrative example of the process. The algorithm constitutes two phases (main steps). The first phase aims to extract and group candidate terms being potentially relevant to each entity (i.e., as a term). However, the candidate terms are not necessarily relevant to the domain defined by the ontology (terms can be ambiguous). Consequently, the aim of the last phase is to identify those groups of candidate terms being most relevant to the entities defined by the ontology. Finally, an FV for each entity is created based on the most prominent group of candidate terms for each entity. The result of this algorithm is a list of entities with corresponding FVs that consist of terms associated with both the entities and the domain terminology.

Figure 2 The Feature Vector Construction algorithm with illustrative example.



The FVC algorithm is designed to be flexible in the sense that it can be tailored to the intended usage of the FVs as well as the different quality of the ontologies. Consequently, the algorithm provides several options at each step. Below, we elaborate each of the steps as follows.

Step 1: Search and cluster

This step constitutes four sub-steps where the aim is to extract candidate terms that are relevant to each entity (i.e., the entity as a term and not the entity as a concept at this stage). The candidate terms are grouped and then, in Step 2, further processed to identify which of the candidate groups being most relevant to the domain of interest defined by the ontology.

Step 1.1: Compose entity query

In this step, a search query is prepared for each entity while the actual search is performed in Step 1.2. The query is based on the entity label with an option to include relevant neighbouring entities and/or keyword(s). Here we aim at creating a query that reflects on the ontology (or a relevant part of it, see (Bhatt et al., 2006)) by considering the closest neighbours of a particular entity, i.e., *parents*, *children*, and *other* entities (Tomassen & Strasunskas, 2009a).

Larger ontologies tend to include several minor domains. By experimentation we found that for diverse ontologies, like the Wine ontology (presented in section 4.2) that also imports a Food ontology, it can be beneficial to add keyword(s) that represents the overall subject domain. The result of using keyword(s) is less distinct and more homogeneous FVs. On other hand, omitting keywords would create FVs that are more distinct and true to the local variances in the ontology.

Step 1.2: Entity based search

The query for each entity created in Step 1.1 is used to retrieve candidate documents for each entity. Any search engine can be used in this step. Currently, Yahoo!® and Google® (for searching in Web documents) and Nutch™ (for searching in local documents) are supported. The user interface is keyword-based.

Step 1.3: Contextual key-phrase extraction

For each document, a set of key-phrases and keywords is extracted, hereinafter referred to as key-phrases. First, a part-of-speech (POS) tagger is used to tag the retrieved documents (snippet or full text). Then a set of tagging rules is applied and a set of candidate noun key-phrases are extracted. Each key-phrase is stemmed to remove duplicates. Finally, those candidate key-phrases above a specified frequency threshold (dependent on the document length) are kept and stored in a document feature vector (DFV) of the corresponding document.

Step 1.4: Cluster search results

In order to identify (discriminate) different subject domains within the documents found for each entity, clustering techniques are used. Recall that the retrieval session is keyword-based (Step 1.2), consequently the terms (entities) can be part of many domains. Clustering allows finding these domains. The Lingo algorithm, from the Carrot2 API (Carrot2, 2009), is used since it performs well for both snippets and full-text documents. The result of this step is a set of clusters for each entity. In addition, for each cluster a cluster feature vector (CLFV) is created. A CLFV is a combination of all the DFVs of a cluster. In the following step, we deal with selecting the relevant cluster w.r.t. the domain of interest.

Step 2: Identify domain and create FV

This step constitutes two sub-steps, aiming to identifying the most relevant clusters w.r.t. the ontology.

Step 2.1: Identify domain relevant clusters

A problem at this stage is to identify the correct subject domain, that is, the most relevant clusters found in Step 1.4 w.r.t. the ontology. Therefore, we compute the similarity between the cluster feature vectors of an entity with the CLFVs of the neighbouring entities, i.e., *parents*, *children*, and *other* entities. In order to find the most prominent cluster, an entity must have at least one neighbour otherwise this check will fail.

Commonality, i.e., high similarity, identifies the document sets or clusters being most relevant to the domain of our interest (defined by the ontology). The hypothesis is that individual clusters having high similarity with neighbouring entities are with high probability of the same domain. This hypothesis is backed up by observed patterns of collocated terms within a domain, equally different domains have different collocation pattern of terms. However, the similarity of clusters depends a lot on the quality of the ontologies, especially the semantic distance between the entities. The result of this step is a domain relevance score for each cluster of an entity with respect to the ontology.

Step 2.2: Construct feature vector

The cluster with the highest domain relevance score, calculated in Step 2.1, is selected for each entity. The step of creating the final FV for the selected cluster can either be based on the already created CLFV of the selected cluster (Step 1.4) or a deeper analysis of the cluster's documents. In the experiments described in section 4, the CLFVs were used.

Figure 1 depicts an illustration of the relationships between a set of documents, an entity, and a FV created using the algorithm presented above.

4 Experiment

In this section, we present the experiment conducted to validate the proposed feature vector quality measures. In section 4.1, we provide an overview of the experiment. Then, the evaluation measures are presented in section 4.2. In section 4.3, we describe how the entities were selected for this experiment. Finally, in section 4.4, the ontologies are described.

4.1 Experiment setting

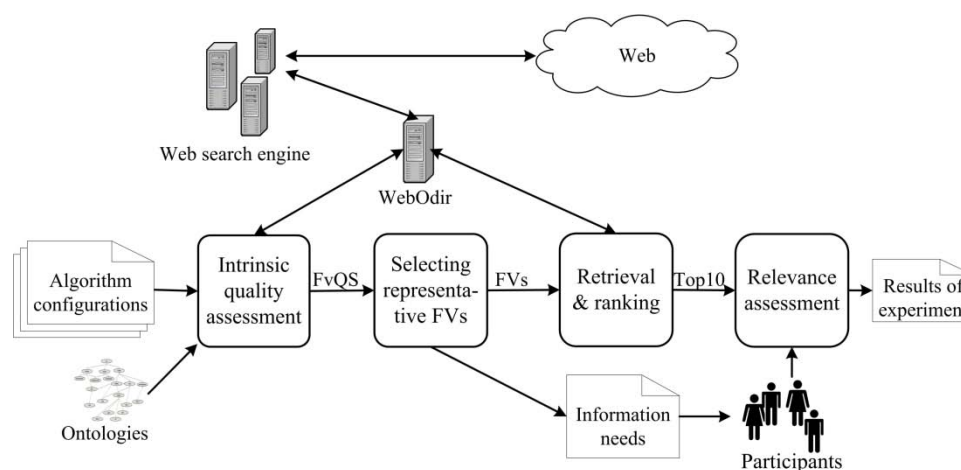
The participants of our experiment were mainly colleagues at the Norwegian University of Science and Technology (NTNU). Nine subjects took part in the experiment. They were not offered any form of compensation for their used time; instead, an amount of money was donated to the Red Cross, an international humanitarian relief agency, for each of the participants.

The design of the experiment is elaborated in Figure 3. A set of ontologies (presented in section 4.4) was populated with different algorithm configurations (described in section 4.1). Next, the quality of the created FVs, with respect to both the ontology and the Web, was assessed. Then, we selected a set of entities that best matched the selection criteria's described in section 4.3. Based on these selected entities

a set of information needs was specified for each of the entities (the simulated information needs are provided in section 5.1.3). In addition, a query was formulated and submitted to our semantic search system (more details of the semantic search system is found in (Tomassen & Strasunskas, 2009b)). A set of queries were formed based on the labels of the selected entities. The top ten results for each query were recorded and presented to the users to evaluate.

The participants of the experiment were presented three simulated information needs with corresponding search results retrieved from our semantic search system. Each query was submitted three times to the semantic search system, but using different FVs that were created as result of different parameters (i.e., LQP, MQP, and HQP settings presented in section 4.3). In total nine queries were submitted and nine evaluation pages were generated each having 10 top ranked documents. Consequently, each user needed to evaluate 90 retrieved documents.

Figure 3 Design of the experiment.



The Yahoo!® Web Search API was chosen as the backend search engine when populating the ontologies. Ideally, the Text Retrieval Conference (TREC) data should be used as baseline. However, we experienced the same problems as d'Aquin et al. (2008) in finding good ontologies that covered TREC. In fact, d'Aquin et al. found that those ontologies available on the Web covered only 20 percent of the domains described in TREC (they used the 100 queries from the WT10G test collection). Google® was chosen to assess the quality of the FVs w.r.t. the ontologies while Yahoo!® was chosen to create the FVs. The reason for this mix of search engines is the limitations of their API's. Currently, Google has a limitation of retrieving maximum 64 documents per search, but an unlimited number of queries can be submitted per day (Google, 2009). Yahoo!®, on the other hand, has an unlimited number of documents that can be retrieved per search results, but a maximum of 5000 queries can be submitted per day (Yahoo, 2009). Since the top 100 retrieved documents per search result are used when constructing the FVs, Yahoo!® had to be used in this process. Since more than 20.000 queries were submitted to assess the quality of the FVs (see section 5.1), Google® had to be used. This is not an ideal scenario, which we address when we discuss the validity of the findings in section 5.3.

4.2 Evaluation measures

In this section, we present the evaluations measures used to evaluate the quality of the constructed feature vectors and the semantic search approach. The quality of the FVs is considered using both intrinsic and extrinsic measures with respect to the ontologies used. The latter evaluation measure is using the Web. Alternatively, real users could assess the FV quality. However, this would not be a practical solution considering a scenario with many or larger ontologies. In addition, such approach would be vulnerable to different interpretations by each individual user. Therefore, we have proposed a more practical and neutral approach to assess the FV quality in the next subsection (introduced in (Tomassen & Strasunskas, 2010)). The quality of the search results from the semantic search approach is evaluated using real users, providing subjective indications of the search result quality. First, the measures used to assess the quality

of the FVs are presented, and then the measure used to indicate the quality of the semantic search approach is presented.

4.2.1 FV quality evaluation measures

In this section, we present the feature vector quality measures proposed in (Tomassen & Strasunskas, 2010) but presented here as well for the sake of easiness. In total, four measures have been defined. The Average FV Similarity (AFvS) and Average FV Neighbourhood Similarity (AFvNS) are both intrinsic measures indicating the uniqueness and the neighbourhood similarity aspects of the FVs. While the Average FV NGD (AFvNGD) is an extrinsic measure used to find the semantic distance between the entities and their FVs. Finally, the Average FV Quality Score (AFvQS) provides a total score by being an aggregated score of the above three measures. These scores give a representative value of the FV quality with respect to the ontologies.

First, the Average FV Similarity (AFvS) is defined. AFvS gives an indication of the uniqueness of the FVs and is defined as follows.

Definition 1: Average FV Similarity (AFvS)

$$AFvS(o) = \frac{2}{n^2 - n} \sum_{i=1}^n \sum_{j=i+1}^n sim(fv_i, fv_j)$$

where n is the number of fvs in the ontology o and $sim(fv_i, fv_j)$ is a similarity between the two vectors. A score of zero indicate that all FVs are unique. In general, we would like this score to be as low as possible in order to discriminate the FVs, but this depends a lot on the quality of the ontology.

The Average FV Neighbourhood Similarity (AFvNS) score indicates the degree of overlap with neighbouring entities and is defined as follows.

Definition 2: Average FV Neighbourhood Similarity (AFvNS)

$$AFvNS(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m sim(fv_i, fv_j)$$

where n is the number of fvs in the ontology o and m is the number of neighbouring entities with fvs of entity i with fv_i . The range is $[0, \dots, 1]$. Note that $AFvS \leq AFvNS$, and, as for AFvS, AFvNS is highly dependent upon the ontology quality.

Normalized Google Distance (NGD) (Cilibrasi & Vitanyi, 2007) was used in the Average FV NGD (AFvNGD) score. The AFvNGD score indicates the semantic distance between the entities and their FVs and is defined as follows.

Definition 3: Average FV NGD (AFvNGD)

$$AFvNGD(o) = \frac{1}{n} \sum_{i=1}^n \frac{1}{m} \sum_{j=1}^m NGD(fv_i, kp_j)$$

where n is the number of fvs in the ontology o and m is the length of the fv_i and fv_i is the name of the fv_i , the entity name, and kp_j are the key-phrases of fv_i . Note, if an entity has a parent, then the name of the parent is also included to provide a more specific similarity distance (adapted from (Bouquet et al., 2003) that in our case is limited to the closest parent). $AFvNGD(fv)$ will have a score in the same range as NGD, that is, $[0, \dots, \infty]$ where zero indicates the best match. However, in practice, most values are in the range of $[0, \dots, 1]$. Consequently, for the special case where $NGD(fv_i, kp_j) > 1$ we set $NGD(fv_i, kp_j) = 1$. The motivation behind this is that the distance is too large to be of any interest anyway and hence we omit values above one.

Once AFvS, AFvNS, and AFvNGD are found, the total score can be calculated. The total score is an aggregated score of the above three measures. The total FV quality score provides the overall quality of the FVs and is defined as follows.

Definition 4: Average FV Quality Score (AFvQS)

$$AFvQS(o) = \alpha(1 - AFvS) + \beta AFvNS + \gamma(1 - AFvNGD)$$

where $\alpha + \beta + \gamma = 1$ are weight factors (defaults are 1/3). The total FV quality score for an ontology will be in the range 0-1, where 1 indicates the best score.

The measures above are believed to provide a representative picture of the quality of the FVs. The weights used in the *Average FV Quality Score* can be tailored to the use of the FVs (i.e. semantic search).

4.2.2 User evaluation measures

In (Strasunskas & Tomassen, 2008) we used a relevance score, adopted from (Brasethvik, 2004), being similar in spirit to Discounted Cumulated Gain (DCG) by Jarvelin & Kekalainen (2002), to evaluate our semantic search approach and hence indirectly the quality of the FVs. Real users marked each of the top 10 retrieved documents according to perceived relevance. This relevance score substitutes conventional precision metric, i.e., precision and recall (Baeza-Yates & Ribeiro-Neto, 1999). We decided to focus on precision instead of recall since we target Web search where precision (i.e., relevant documents at top positions) is more important than recall.

A relevance score for each query was calculated as follows:

$$Score_q = \frac{1}{2} \sum_{i=1}^{10} P_{D_i} \times P_{P_i}$$

where P_{D_i} is an individual score for document D_i , and P_{P_i} - the weighting factor for position P_i . Score for document is as follows: -1 for trash; 0 for non-relevant or duplicate; 1 - related; and 2 - good document. Document ranking position has weights as follows: 1st - 20; 2nd - 15; 3rd - 13; 4th - 11; 5th - 9; 6th & 7th - 8; 8th & 9th - 6; 10th - 4. Consequently, the final score falls into a range [-50, 100].

4.3 Entity selection

In this section, we describe the parameters used to populate the ontologies (presented in section 4.1) using the feature vector construction approach (presented in section 3.2) and the motivation behind them. Since the goal of this paper is to validate the evaluation measures (described in section 4.2.1) a set of best practice parameters from earlier experiments (Tomassen & Strasunskas, 2010) were used in this paper. Further, the motivation for using these best practice parameters was to get a set of distinct FVs to better assess the effect of the FVs and hence validate the correlated FV quality scores (presented in section 4.2.1). Based on earlier experiments we found that the FV quality can be classified into low, medium, and high. Therefore, three sets of parameters were used: Low Quality Parameters (LQP), Medium Quality Parameters (MQP), and High Quality Parameters (HQP). The parameters are described as follows:

Low Quality Parameters (LQP): Based on lessons learned from earlier experiments (Tomassen & Strasunskas, 2010) a set of parameters expected to provide low quality FVs were selected. Consequently, we used as little knowledge as possible from the ontologies in the construction process since earlier experiments have shown that richer ontologies provide better results (Strasunskas & Tomassen, 2008). Similarly, the queries were not expanded and hence provided more ambiguous queries that resulted in FVs with more noise. Equally, few neighbouring entities were used to identify the most relevant cluster with respect to the ontology (see Step 3.1 in section 3.2 for more details). These parameters are in general expected to create FVs with relatively low quality.

Medium Quality Parameters (MQP): A set of expected medium quality parameters was also defined. In contrast to the LQPs, the queries were expanded with neighbouring entities, i.e., parental entities. In addition, more neighbouring entities were used to identify the most relevant cluster with respect to the ontology. In general, these parameters were expected to create FVs with higher quality than those created with the Low Quality Parameters but with lower quality than those created with the High Quality Parameters presented next.

High Quality Parameters (HQP): Finally, we created a set of parameters that was expected to provide FVs of high quality. A semantic reasoner is utilised, i.e., the Pellet OWL Reasoner (Sirin et al., 2007), to

extract all the relationships among the entities. A reasoner can affect both the number of relations for each entity and the number of entities being available. For example, if two classes are sub classes of `owl:Thing` and equivalent, then all their properties and sub hierarchical structure should be equal. These "additional" relations are found by a semantic reasoner. In addition, even more neighbouring entities are included to both expand the queries and to identify the most relevant clusters with respect to the ontologies. These parameters were in general expected to create FVs with higher quality than those created with the Medium Quality Parameters.

The three sets of parameters defined above are summarized in Table 1. The result of applying these parameters to the feature vector construction algorithm are ontologies with associated FVs of different quality (i.e., low, medium, and high). The results and the analysis of these are given in section 5.

To validate the FV quality measures described in section 4.2.1, representative entities need to be identified and selected. These selected entities, will form as the basis for the queries submitted to our semantic search system and next evaluated by real users (recall section 4.1). Consequently, we needed an approach to identify those entities for each of the ontologies that best reflected the differences of the selected FV construction parameters described above. Therefore, two equations were defined that measured the most equal (i.e., Distribution Equality (DE)) and the largest span (i.e., Distribution Span (DS)) between the FVs with respect to the measured FV quality. The motivation behind these measures was to find those FVs that best reflected the parameters summarised in Table 1, that is, low, medium, and high. Therefore, it was assumed that the best FV candidates were those with the largest possible gap between LQP and HQP and where MQP was as centric between those scores as possible. Therefore, we defined in addition a Total Distribution Score (TDS), i.e., an aggregation of the DE and the DS measures that reflects the best FV candidates.

Table 1 Summary of quality parameters used to construct the FVs.

	<i>Low Quality Parameters</i>	<i>Medium Quality Parameters</i>	<i>High Quality Parameters</i>
<i>Ontology analysis</i>			
With reasoner			X
Without reasoner	X	X	
<i>Query expansion</i>			
Parents		X	X
Children			X
Others			X
<i>Search results</i>			
Number of results	100	100	100
<i>Domain identification</i>			
Parents	X	X	X
Children		X	X
Others			X

Since the objective of this paper is to validate our proposed metrics (presented in section 4.2.1), the metrics were also used to measure the quality of the FVs and hence served as the basis for the entity selection. However, the intention of the proposed metrics was to provide an overall score of the FV quality with respect to an ontology. To select a representative entity w.r.t. both the parameters presented above and the ontologies used we need to focus on each specific FV. Therefore, the AFvS measure was omitted because of its nature (i.e., considering the populated FVs from an overall view). While the AFvNS and the AFvNGD measures were adapted to reflect the individual FVs by omitting the average part of these measures (i.e., $AFvNS(o)$ was altered to $FvNS(fv)$ instead and likewise for AFvNGD). The adapted measures were denoted as FvNS and FvNGD respectively. Equally, the AFvQS was adapted by omitting the AFvS score and the average part of the equation omitted (i.e., $AFvQS(o)$ was altered to $FvQS(fv)$), and hence been denoted as FvQS. Further, the weights used by FvQS (i.e. β and γ) are equally

distributed as the default weights and hence set to 1/2 for each to make the measurements comparable to our previous work presented in (Tomassen & Strasunskas, 2010).

The Distribution Equality (DE), Distribution Span (DS), and Total Distribution Score (TDS) measures are defined as follows:

Distribution Equality (DE): The equality between $|FvQS_{Low} - FvQS_{Med}|$ and $|FvQS_{Med} - FvQS_{High}|$. Let $e \in \{E/O\}$, e is an entity in the ontology O . E/O is the set of entities in ontology O .

$$DE(e) = 1 - \frac{|FvQS_{High}(e) - 2 \times FvQS_{Med}(e) + FvQS_{Low}(e)|}{2}$$

where $DE(e)$ is the Distribution Equality for an entity e . The score is a value between $[0..1]$, where 1 is the best score.

Distribution Span (DS): The span between the lowest score $FvQS_{Low}$ and the highest $FvQS_{High}$. Let $e \in \{E/O\}$, e is an entity in the ontology O . E/O is the set of entities in ontology O .

$$DS(e) = FvQS_{High}(e) - FvQS_{Low}(e)$$

where $DS(e)$ is the Distribution Span for an entity e . The score is a value between $[0..1]$, where 1 is the best score.

Total Distribution Score (TDS): This is an aggregate score of DE and DS. Let $e \in \{E/O\}$.

$$TDS(e) = \frac{DE(e) + DS(e)}{2}$$

where $TDS(e)$ is the Total Distribution Score for an entity e .

Note, the prerequisites are that $FvQS_{Low}(e) \leq FvQS_{Med}(e) \leq FvQS_{High}(e)$ and that $FvQS_{Low}(e) > 0$.

Based on these criteria a set of entities were selected (listed in section 5.1.2).

4.4 Ontologies

The same set of ontologies used in the experiments (Tomassen & Strasunskas, 2009b) was selected for the experiments presented in this paper since we revisit our approach to construction of semantic-linguistic feature vectors. The ontologies are of different granularity and are formalized in OWL. The ontologies are as follows:

Animals ontology: A small ontology that classifies some species, does not contain any individuals, and has only hierarchical properties. The ontology was selected to see the effect of applying the approach on a typical taxonomy.

Travel ontology: A bit more advanced, compared to the Animals ontology, by having in addition both individuals and some object properties. This ontology is classified in this work as a lightweight ontology.

Wine ontology: Even more advance than the Travel ontology with more individuals than classes and many relations. This ontology was originally constructed to test reasoning capabilities. Maybe as a result, the ontology contains some entity labels not found elsewhere (e.g., the entity McGuinnesso is according to the ontology a winery; however, a search with Google provides no results). Consequently, several entities will not be populated with this ontology. This ontology is classified in this work as advanced.

The key characteristics of the ontologies are displayed in Table 2 (the ontologies can be accessed at <http://research.idi.ntnu.no/IIP/ontologies/>).

Table 2 Ontology key characteristics.

<i>Ontology</i>	<i>Classes</i>	<i>Individuals</i>	<i>Properties</i>
Animals	51	0	0
Travel	34	14	6
Wine	82	155	10

In this evaluation, we did not focus on performance issues like processing time, scalability, etc. Therefore, we did not include any large or heavyweight ontologies since we believe that larger ontologies will not provide any significant new insights except of processing time.

5 Results and analysis

In this section, we present the results and analyze the data collected during the experiment described in section 4. First, we present the results of the experiment and how the entities were selected and then we analyse the results. In section 5.3, we discuss the threats to the validity of the results, before we summarise.

5.1 Results

The different quality parameters (summarized in Table 1) were applied when populating the ontologies resulting in nine different configurations. The experiments were performed on a standard PC with an Intel® Pentium processor running Windows™ XP, running Apache Tomcat. Populating and analyzing the ontologies took more than 10 hours; the most complex ontology, the Wine ontology, took from 133 to 197 minutes to populate and analyse. When populating the ontologies and evaluating the quality of the FVs, more than 20.000 queries were submitted to the Google®.

5.1.1 Feature vector quality

The ontologies were populated using different quality parameters. The quality of the feature vectors was assessed with the described measures (section 4.2.1 with the adaption described in section 4.3). Table 3 summarises the test results. Since we in this experiment focus on each individual entity in contrast to the ontologies as whole, only the best-matched entities, according to the criteria presented in section 4.3, are shown. The best (in **bold**) and least (in *italic*) scores for each ontology with respect to the different parameters are highlighted. Note, that the best score for FvNGD is a score close to zero.

Note, that "Bunjee Jumping" is misspelled in the original Travel ontology (found on the Web) being used in this experiment. We have selected not to fix such faults in the ontologies and, therefore, "Bunjee Jumping" is used throughout in this paper.

Table 3 FV quality scores (bold indicates the best values while italic the least).

<i>Ontology</i>	<i>Entity</i>	FvNS			FvNGD			FvQS		
		<i>Low</i>	<i>Med</i>	<i>High</i>	<i>Low</i>	<i>Med</i>	<i>High</i>	<i>Low</i>	<i>Med</i>	<i>High</i>
Animals	Hare	<i>0,251</i>	0,609	0,911	<i>0,283</i>	0,244	0,240	<i>0,670</i>	0,741	0,775
Travel	Bunjee Jumping	<i>0,014</i>	0,200	0,570	<i>0,130</i>	0,107	0,105	<i>0,784</i>	0,824	0,862
Wine	Dessert Wine	<i>0,642</i>	0,763	0,911	<i>0,158</i>	0,151	0,130	<i>0,822</i>	0,841	0,874

5.1.2 Entity selection

Table 3 shows the best-matched entity for each of the ontologies. The selection of these was done using the criteria described in section 4.3. The Distribution Span, Distribution Equality, and the Total Distribution Score are summarized and shown in Figure 4. The figure only shows those FVs that best reflected the set of parameters used.

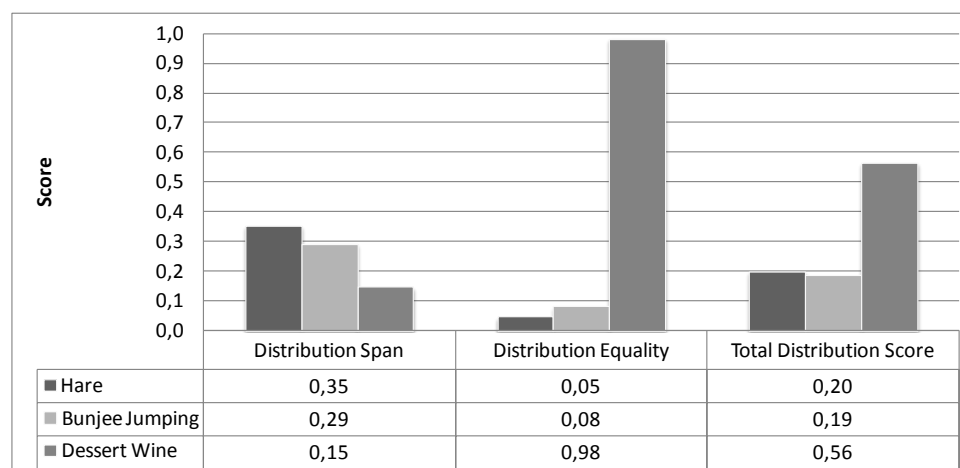
Figure 4 Entity selection scores.

Table 4 depicts an example of FVs for the Bunjee Jumping entity where different quality parameters were used. As can be seen from the table, even though Bunjee is misspelled, Bungee (the correct form) is included in the FVs as well. Note, that the misspelling of Bunjee Jumping influenced the number of search results (i.e., Yahoo!®, as of 15th of March 2010, returned 47.900 results for the Bunjee Jumping query in contrast to 1.220.000 for Bungee Jumping).

Table 4 Examples of FVs for the Bunjee Jumping entity created with different parameters.

With low parameters	Weight	With medium parameters	Weight	With high parameters	Weight
bunjee	1,0	bridge	1,0	bunjee	1,0
bungee	0,8	bunjee	0,8	adventure	0,7
bungee jump	0,6	bungee	0,7	bungee	0,6
world	0,6	adventure	0,5	bunjee jump	0,3
commercial bungee	0,4	archery bunjee	0,3	adventure sports	0,2
commercial bungee jump	0,4	trails	0,3	birth place	0,2
highest commercial bungee	0,4	air	0,2	free online	0,2
tower	0,4	been	0,2	meet	0,2
african tourism owns	0,2	bhote kosi	0,2	other	0,2
black water	0,2	bridge bunjee	0,2	other adventure sports	0,2
bridge south africa watch	0,2	caravans	0,2	other people	0,2
bunjee jump	0,2	choose from	0,2	put away your credit	0,2
copyright	0,2	choose from our	0,2	reverse	0,2
day	0,2	company	0,2	singles	0,2
eastern cape province south	0,2	down from	0,2	site put away your	0,2
hello you either have	0,2	elephant back safari helicopter	0,2	adventures tourism bungee	0,1
images	0,2	forest	0,2	aerial stunts base	0,1
lyell	0,2	indoor bungee	0,2	agency offer adventure travel	0,1
old	0,2	just imagination	0,2	always	0,1
option	0,2	meters	0,2	experience fun happiness	0,1
orlando towers	0,2	options	0,2	insurance single trips	0,1
photos eastern cape	0,2	our many bungee	0,2	joe jennings	0,1
province south africa route	0,2	outdoor adventure	0,2	justsayhi our	0,1
rural bunjee	0,2	people have	0,2	mountain	0,1
travel	0,2	safari par excellence	0,2	offers outdoor adventure	0,1
vertical adventure center	0,2	self-drive safaris since	0,2	pyrenees	0,1
video	0,2	someone jumps	0,2	reverse bunjee jump	0,1
window bunjee	0,2	themselves	0,2	rock	0,1
world through photos	0,2	tropical gorge	0,2	sports like whitewater	0,1
you either have javascript	0,2	videosu klibi izle indir	0,2	tragedy adventure sports	0,1

5.1.3 Simulated information needs

Based on the selected entities (Figure 4), according to the criteria defined in section 4.3, a set of simulated information needs were created. The simulated information needs are depicted in Table 5. The information needs were designed to be fairly basic and general since only the labels of the selected entities were used to form the queries submitted to the semantic search engine. Consequently, the information needs were designed to reflect the corresponding generic queries.

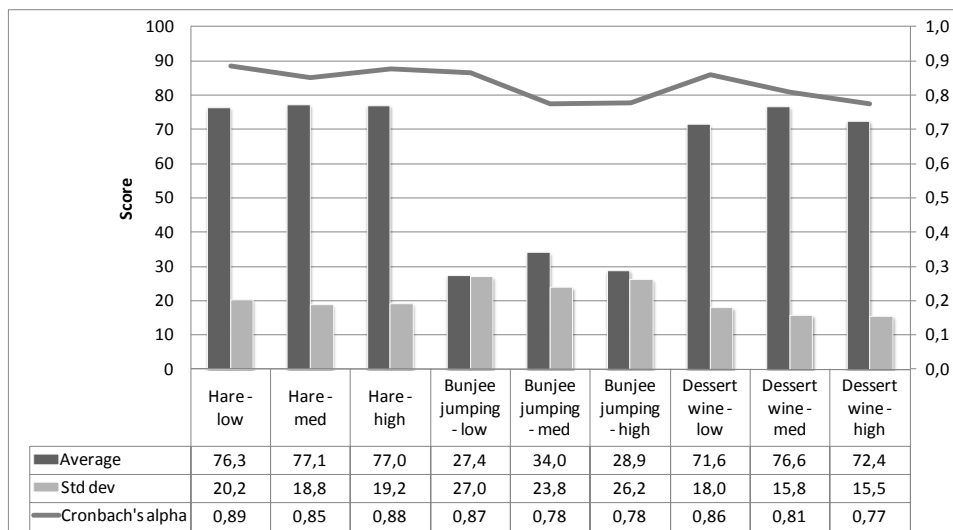
Table 5 Simulated information needs.

Query	Simulated information need
Hare	Find some basic information about a hare (an animal). The information needed is basic, that is, to find out what a hare is.
Bunjee Jumping	This time you are interested in adventure and looking for information about an activity or sport. What is bungee jumping?
Dessert Wine	Finally you want to improve your knowledge about dessert wines. Which wines are dessert wines?

5.2 Analysis

In this section, the results of the experiment are analysed. Figure 5 depicts an overview of the search result relevance scores (i.e., the average of the scores) along with the standard deviation and the Cronbach's alpha scores. As can be seen, the Cronbach's alpha scores are above 0,7 for all the evaluations with an average 0,8. A Cronbach's alpha score above 0,7 indicates that the results of the evaluation is reliable.

Figure 5 Search result relevance score (range [-50, 100]) and Cronbach's alpha for selected entities.

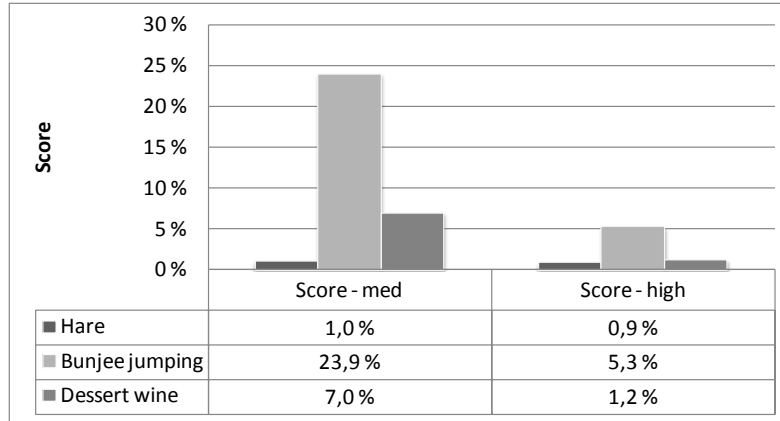


Further, we observed that the average score for all the evaluated queries were above zero (i.e., an average score below zero would indicate that the documents are perceived as either thrash or irrelevant). However, the score is considerable lower for the Bunjee Jumping entity than the Hare or the Dessert Wine entities. This might be explained by the fact that Bunjee Jumping is misspelled and, consequently, results in less relevant results retrieved. The standard deviation is also higher for this entity, indicating an uncertainty among the users about the relevance of the retrieved documents.

From Figure 5, we see the lowest average relevance scores in all the cases where the Low Quality Parameters (see section 4.3) were used to construct the feature vectors. Further, we observed that the highest scores were achieved for all the cases where the Medium Quality Parameters were used. Figure 6 depicts the relevance scores for the entities used with the MQP and the HQP parameters relative to the

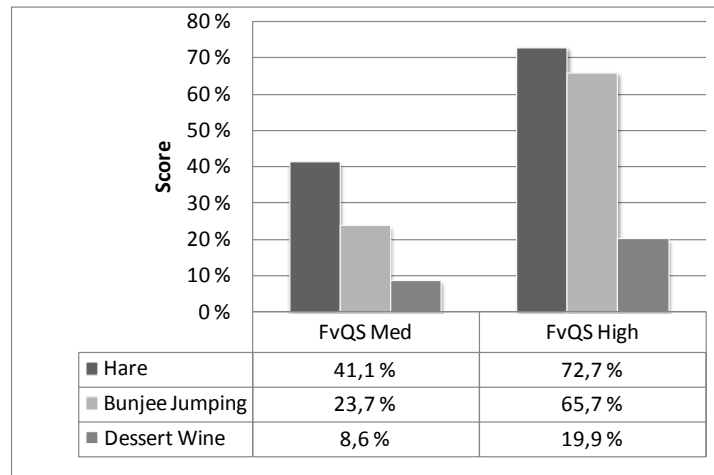
entities where the LQP were used in percentage. We observed the same pattern; in cases where the FVs were created using the MQP the score was highest.

Figure 6 Search result relevance score relative to the lowest scores.



Next, we analysed how well these results matched the assessed quality of the FVs using the proposed evaluation measures. Figure 7 depicts the assessed FV quality scores for the entities used with the MQP and the HQP parameters relative to the entities where the LQP were used in percentage. As can be observed, the FVs constructed with the High Quality Parameters provided the best scores. However, the FVs assessed to have the highest score did not provide the best scores when used in search (see Figure 6 versus Figure 7). The best scores were achieved when the FVs were created using the Medium Quality Parameters. However, we observed that the FVs created using the Low Quality Parameters provided the lowest score both for the assessed FvQS and when used in search (see Table 3 versus Figure 5). Consequently, we concluded that FVs created using the MQP and the HQP provided better FVs for use in search than FVs created with LQP. However, HQP is not necessarily the best parameters for creating FVs to be used in search.

Figure 7 Top 1 FV quality scores relative to the lowest score.



Recall that the Average FV Quality Score (AFvQS) is an aggregated score of the Average FV Similarity (AFvS), Average FV Neighbourhood Similarity (AFvNS), and the Average FV NGD (AFvNGD) scores. AFvS, AFvNS, and AFvNGD can be tuned using the α , β , and γ weights. However, because of the way the different scores are aggregated, i.e., being linear, AFvQS cannot be tuned to fit the observations done in this experiment. Consequently, AFvQS needs to be revised to better reflect FVs used in search.

5.3 Threats to validity

Possible threats to the results and the analysis presented in above are as follows.

- External validity describes a degree to which the results can be generalized outside the experiment. The experiment was conducted using only one system (the prototype implementation of the outlined approach). However, the conclusions and lessons learned are applicable to all similar approaches, especially ones using ontologies to construct FVs to be applied in a search context.
- The case study was executed at a university. However, since the users only evaluated the provided search results and were not able to influence on the submitted query it is believed that individual computer skills was of minor importance.
- Users provided subjective evaluations. The individuals were given simulated information needs and were not able to influence on the submitted query. A set of entities were selected, based on a set of criteria, which formed as a basis for the queries. The simulated information needs were designed to reflect the queries in a general manner. In this experiment, we observed the difference between the results of each individual query for each information need (i.e., three queries with different parameters used for each information need). Based on an information need a user might have liked to submit an alternative query than used. However, if a user disagrees with a submitted query that negativism would apply to all three search results for that information need. Consequently, it is believed to be of minor importance if the submitted query was not optimal according to the users preferences since that would apply to all the queries for each simulated information need. Moreover, the Cronbach's alpha scores were above 0,7 indicating reliable evaluation results.
- The selection of entities was done using a set of defined criteria. However, the assessed FV quality was done using the proposed evaluation measures with the same weights used in earlier experiments. Further, the selection of the FVs was based on criteria that best reflected the diversity of the parameters used to get good observations (i.e., distance between the results to better observe trends).
- Fatigue effect. On average, half an hour were spent to evaluate the ninety pages. Therefore, this effect is not considered relevant.
- Use of different search engines to construct the FVs and assess the quality of the FVs. In (Tomassen & Strasunskas, 2010) we found that change of comparable search engines does not yield any major effect on the FV quality if an adequate number of search results is used. In this paper, the top 100 retrieved documents were used to construct the FVs, which are considered as a sufficient amount of search results.

6 Conclusions and future work

In this study, we have described and evaluated an unsupervised approach to feature vector construction. The proposal is based on a non-supervised solution that is applicable to any ontology as long as there is some correlation between the ontology and the text corpus. We provided an overall description of the process of associating each entity of an ontology with a FV.

In the evaluation, we investigated the applicability of the proposed metrics for assessing FV quality. Ontologies of different granularity have been used and populated using three different configurations. The quality of the associated FVs has been assessed. A set of selected entities was used to provide a set of search results that were evaluated by real users. The assessed quality of the FVs was compared against the assessed quality of corresponding search results. Findings show, that ontologies populated using the defined Medium Quality Parameters provided the best results and that the Low Quality Parameters provided the lowest scores. These results indicate that our proposed metrics provide in general good indications of the FV quality.

A limited number of ontologies were used in this experiment. Therefore, one of the future tasks is to conduct a similar experiment with more ontologies. Further, we need to investigate alternative approaches to aggregate the total feature vector quality score to better reflect the needs of search applications and different search tasks too. Those are the main future tasks.

Acknowledgements

This research work is partially funded by the Information Access Distributions (iAD) project that is partially funded by the Research Council of Norway (NFR) as a Centre for Research-based Innovation (SFI).

References

- Agirre, E., Ansa, O., Hovy, E.H. & Martínez, D. 2000, 'Enriching very large ontologies using the WWW', *ECAI Workshop on Ontology Learning*, vol. 31, viewed 16.02.2010 <<http://CEUR-WS.org/Vol-31/>>.
- Alemayehu, N. (2003) 'Analysis of performance variation using query expansion', *J. Am. Soc. Inf. Sci. Technol.*, Vol. 54, No. 5, pp. 379-391.
- Baeza-Yates, R. & Ribeiro-Neto, B. (1999) *Modern information retrieval*, ACM Press, New York.
- Bhatt, M., Flahive, A., Wouters, C., Rahayu, W. & Taniar, D. (2006) 'MOVE: A Distributed Framework for Materialized Ontology View Extraction', *Algorithmica*, Vol. 45, No. 3, pp. 457-481.
- Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D. & Dillon, T. (2004) 'A Distributed Approach to Sub-Ontology Extraction', paper presented to the *Proceedings of the 18th International Conference on Advanced Information Networking and Applications - Volume 2*.
- Bhogal, J., Macfarlane, A. & Smith, P. (2007) 'A review of ontology based query expansion', *Inf. Process. Manage.*, Vol. 43, No. 4, pp. 866-886.
- Bouquet, P., Serafini, L. & Zanobini, S. (2003) 'Semantic Coordination: A New Approach and an Application', in *The Semantic Web - ISWC 2003*, Vol. 2870, Springer, Heidelberg, pp. 130-145.
- Brasethvik, T. (2004) 'Conceptual modeling for domain specific document description and retrieval - An approach to semantic document modeling', NTNU, Trondheim.
- Bry, F., Koch, C., Furche, T., Schaffert, S., Badea, L. & Berger, S. (2005) 'Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages', *Int. J. Semantic Web Inf. Syst.*, Vol. 1, No. 2, pp. 1-21.
- Carmel, D., Yom-Tov, E., Darlow, A. & Pelleg, D. (2006) 'What makes a query difficult?', paper presented to the *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, USA.
- Carrot2 (2009), *Carrot2 an Open Source Search Results Clustering Engine*, <<http://www.carrot2.org/>>.
- Castells, P., Fernandez, M. & Vallet, D. (2007) 'An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 2, pp. 261-272.
- Cilibrasi, R. & Vitanyi, P. (2007) 'The Google Similarity Distance', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 3, pp. 370-383.
- d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V. & Guidi, D. (2008) 'Toward a New Generation of Semantic Web Applications', *IEEE Intelligent Systems*, Vol. 23, No. 3, pp. 20-28.
- Formica, A., Missikoff, M., Pourabbas, E. & Taglino, F. (2008) 'Weighted Ontology for Semantic Search', in R. Meersman & Z. Tari (eds), *On the Move to Meaningful Internet Systems: OTM 2008*, Vol. 5332, Springer, Heidelberg, pp. 1289-1303.
- Gabrilovich, E. & Markovitch, S. (2007) 'Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization', *J. Mach. Learn. Res.*, Vol. 8, pp. 2297-2345.
- Google (2009), *Google AJAX Search API*, <<http://code.google.com/apis/ajaxsearch/>>.
- Horrocks, I. (2007) 'Semantic web: the story so far', paper presented to the *Proceedings of the 2007 international cross-disciplinary conference on Web accessibility (W4A)*, Banff, Canada.
- Jarvelin, K. & Kekalainen, J. (2002) 'Cumulated gain-based evaluation of IR techniques', *ACM Trans. Inf. Syst.*, Vol. 20, No. 4, pp. 422-446.
- Jiang, X. & Tan, A.-H. (2006) 'OntoSearch: A Full-Text Search Engine for the Semantic Web', paper presented to the *Proc. of the 21st National Conf. on Artificial Intelligence*, Boston, Massachusetts.
- Kulkarni, S. & Caragea, D. (2009) 'Towards Bridging the Web and the Semantic Web', in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, 2009. WI-IAT '09*, IEEE Computer Society, Milano, Italy, pp. 667-674.
- Lopez, V., Sabou, M. & Motta, E. (2006) 'PowerMap: Mapping the Real Semantic Web on the Fly', in *The Semantic Web - ISWC 2006*, Vol. 4273, Springer-Verlag, pp. 414-427.
- Mitchell, T.M. (1997) *Machine Learning*, McGraw-Hill, New York.

- Moscato, F., Martino, B.D., Venticinque, S. & Martone, A. (2009) 'OverFA: a collaborative framework for the semantic annotation of documents and websites', *Int. J. Web Grid Services*, Vol. 5, No. 1, pp. 30-45.
- Noah, S.A., Alhadi, A.C. & Zakaria, L.Q. (2005) 'A semantic retrieval of web documents using domain ontology', *Int. J. Web Grid Services*, Vol. 1, No. 2, pp. 151-164.
- Page, L., Brin, S., Motwani, R. & Winograd, T. 1999, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford InfoLab, viewed 16.02.2010 <<http://ilpubs.stanford.edu:8090/422/>>.
- Panagis, Y., Sakkopoulos, E., Garofalakis, J. & Tsakalidis, A. (2006) 'Optimisation mechanism for web search results using topic knowledge', *International Journal of Knowledge and Learning*, Vol. 2, pp. 140-153.
- Scuturici, M., Clech, J., Scuturici, V.-M. & Zighed, D. (2005) 'Topological representation model for image database query', *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 17, No. 1-2, pp. 145-160.
- Sirin, E., Parsia, B., Grau, B., Kalyanpur, A. & Katz, Y. (2007) 'Pellet: A practical OWL-DL reasoner', *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, pp. 51-53.
- Solskinnsbakk, G. & Gulla, J. (2008) 'Ontological Profiles in Enterprise Search', in *Knowledge Engineering: Practice and Patterns*, pp. 302-317.
- Strasunskas, D. & Tomassen, S.L. (2008) 'The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation', *Int. J. Knowledge and Learning*, Vol. 4, No. 4, pp. 398-414.
- Strasunskas, D. & Tomassen, S.L. (2010) 'On Variety of Semantic Search Systems and Their Evaluation Methods', in *The Proceedings of the International Conference on Information Management and Evaluation*, Academic Conferences Publishing, pp. 380-387.
- Su, X. & Gulla, J.A. (2006) 'An information retrieval approach to ontology mapping', *Data & Knowledge Engineering*, Vol. 58, No. 1, pp. 47-69.
- Suomela, S. & Kekalainen, J. (2005) 'Ontology as a Search-Tool: A Study of Real Users' Query Formulation With and Without Conceptual Support', in *Advances in Information Retrieval*, pp. 315-329.
- Tomassen, S.L. & Strasunskas, D. (2009a) 'Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect', in *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 03*, IEEE Computer Society, Washington, pp. 133-138.
- Tomassen, S.L. & Strasunskas, D. (2009b) 'An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks', in G. Kotsis, D. Taniar, E. Pardede & I. Khalil (eds), *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, ACM, pp. 128-136.
- Tomassen, S.L. & Strasunskas, D. (2010) 'Measuring intrinsic quality of semantic search based on Feature Vectors', *Int. J. Metadata, Semantics and Ontologies*, Vol. 5, No. 2, pp. 120-133.
- Wang, H., Zhang, K., Liu, Q., Tran, T. & Yu, Y. (2008) 'Q2Semantic: A Lightweight Keyword Interface to Semantic Search', in pp. 584-598.
- Yahoo (2009), *Yahoo! Developer Network*, <<http://developer.yahoo.com>>.
- Zhang, L., Yu, Y., Zhou, J., Lin, C. & Yang, Y. (2005) 'An enhanced model for searching in semantic portals', paper presented to the *Proceedings of the 14th international conference on World Wide Web*, Chiba, Japan.
- Zhou, X., Hu, X. & Zhang, X. (2007) 'Topic Signature Language Models for Ad hoc Retrieval', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 9, pp. 1276-1287.

P6: An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks

Publication details

Tomassen, S. L. & Strasunskas, D. (2009) An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks. In: Kotsis, G., Taniar, D., Pardede, E. & Khalil, I. (eds.) *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*, ACM.

An ontology-driven approach to Web search: analysis of its sensitivity to ontology quality and search tasks

Stein L. Tomassen

IDI, NTNU

Sem Saelandsvei 7-9,
NO-7491 Trondheim, Norway
+ 47 735 94218

stein.l.tomassen@idi.ntnu.no

Darijus Strasunskas

IOT, NTNU

Alfred Getz veg 3,
NO-7491 Trondheim, Norway
+ 47 735 93659

darijuss@gmail.com

ABSTRACT

An increasing number of recent information retrieval systems makes use of ontologies to help the users to detail queries and to come up with semantic representations of documents. A particular concern here is user-friendliness (usability) and scalability of those approaches for Web search purposes. In this paper, we present an approach where entities in an ontology are associated with domain terminology by feature vectors (FV). A FV reflects the semantic and linguistic neighbourhoods of a particular entity. The semantic neighbourhood is derived from an ontology and is based on related entities and specified properties, while linguistic neighbourhood is based on co-location of terms in a text corpus. Later, during the search process the FVs are used to filter and re-rank the search results of the underlying search engine and thereby increasing the precision of the result.

We elaborate on the approach and describe how the FVs are constructed. Then we report on a conducted evaluation where we analyse the sensitivity of the approach w.r.t. ontology quality and search tasks. Results indicate that the proposed approach and implemented prototype are able to improve the search results of a standard Web search engine. Furthermore, the analysis of the experiment data shows that the level of ontology specification is important for the quality of the FVs.

Categories and Subject Descriptors

H.3.3 [Information Storage And Retrieval]: Information Search and Retrieval - *information filtering, selection process.*

H.3.4 [Information Storage And Retrieval]: Systems and Software - *performance evaluation (efficiency and effectiveness).*

Keywords

ontology, web search, semantic search, ontology quality, feature vector construction.

1. INTRODUCTION

Given broad Web terminology and limited domain terminology used in an ontology, we endeavour to semantically and linguistically extend domain terminology (terms used to name entities in a domain ontology) in order to improve matching between ontology entities and terminology of documents. The approach presented in this paper utilizes ontologies that are automatically adapted to the corpus' terminology by computing a feature vector (FV) for each entity in the ontology. The idea is to associate every entity (classes and instances) with a FV to tailor these entities to the specific terminology used in the text corpus (the Web). Synonyms and conjugations naturally go into such a vector, but we would also like to include related terms tended used in connection with the entity and provide a contextual definition of it. The FVs are later used to filter and re-rank the search results from an underlying search engine before presentation of the final result. We envision our approach to be used in transition from the current Web and the fully-fledged Semantic Web.

Web search is characterized by having focus on retrieving documents, navigating to a particular Web page, or retrieving a piece of wanted information rather than browsing knowledge or answering a question. Employing ontologies to enhance this type of searches requires certain qualities of the ontologies. For instance, subclass hierarchies are considered sufficient for document retrieval while any other ontology specifications (properties and axioms) are required only for knowledge browsing and question answering [10]. However, here we show that ontology

quality improvement, by specifying equivalent and disjoint classes, adding instances, and properties, can significantly improve Web search results.

The objective of this paper is to present the proposed approach, analyse and discuss the results from an experiment. The experiment has been conducted with potential end-users of such systems. The approach and prototype are evaluated by the means of an experiment and a post-task questionnaire. The paper addresses broad evaluation research questions as follows.

RQ1. How sensitive is the approach to ontology quality? Feature vectors are built based on knowledge specified in ontologies, therefore granularity and quality of encoded knowledge has direct impact on the quality of FVs.

RQ2. Is the approach performance indifferent to various search tasks? Various search strategies and information needs typically concern different granularity of required information. For instance, some prefer finding concrete and concise information on a particular topic, while some are interested in exploring a topic either in depth or in breadth.

Consequently, the novelty and contribution of this paper lies in the analytical experiment attempting to deepen understanding how ontology quality and search tasks aspects affect the overall performance of semantic search.

The rest of the paper is structured as follows. First, we briefly review related work. Then we elaborate on the proposed approach to ontology-driven Web search. Next, we describe a conducted experiment where we evaluate the proposed approach and its sensitivity to ontology quality and search tasks. Then the main results are presented followed by a detailed analysis and discussion. Finally, we conclude the paper and outline future work.

2. RELATED WORK

The Web contains vast resources of information. However, the diversity of topics and terminologies makes it difficult to find relevant information. The Semantic Web (SW) is believed to be the successor of the current Web and provides means to tackle some of these issues. The grand idea is to annotate every piece of information with machine-processable semantic descriptions that enable more advanced usage of the information elements, like reasoning among others. Consequently, there are many initiatives

to semantic search. Some are relying on semantic annotations (e.g., [27]); some are enhancing clustering of retrieved documents (e.g., [17]). There are also many efforts devoted to research on improvement of information retrieval (IR) by using SW techniques. Most of these approaches are utilizing ontologies with encoded domain knowledge to improve search (e.g., [2, 4, 21, 26]). In this section section, we will explore related work where SW techniques are used to enhance search. Since we focus on search task fitness in this paper, a brief overview of information needs and search strategies are provided at the end of this section.

2.1 Semantic search

Search systems for the SW can generally be divided into two categories; those searching for SW documents (i.e., documents expressed in a semantic mark-up languages like OWL, RDF, etc) and those using SW techniques to improve search results [7]. The overview provided here is limited to approaches that endeavour improvement of search by SW techniques (for a more extensive overview of SW systems the reader is referred to [7, 13, 20]). Next, we will provide an overview of the most similar approaches to our work.

Many approaches typically enhance traditional vector space model (VSM) by adding processing of semantics. Nagypal [15] combines ontology usage with the VSM by extending a non-ontological query. There, ontology is used to disambiguate queries. Text search is run on the concepts' labels and users are asked to choose the proper term interpretation. Paralic & Kostial [18] describe a similar approach where documents are associated with concepts in the ontology. The concepts in the query are matched to the concepts of the ontology in order to retrieve terms and then used for calculation of document similarity.

OntoSearch by Jiang & Tan [12] is a full text search engine that depends on documents annotated with elements from an ontology. The user submits a traditional keyword-based query that yields a set of documents. These retrieved documents contain semantic annotations that are used by the spreading activation algorithm to retrieve additional documents and finally rank the documents. Results show that the approach performs better than a compared keyword-based approach.

Formica et al. in [8] proposes a novel way of ranking annotated documents with respect to both

an ontology and a user query. In advance, the documents have been annotated with a set of characterizing concepts, called a feature vector, which they assume already have been built. These FVs function as instances of the corresponding concepts. Similarity between the concepts of a user query and the FVs with respect to the ontology are calculated. Testing shows that their approach performs slightly better than other compared approaches. However, a limitation with the approach is that only the hierarchical structure of the ontology is used when calculating the similarity scores.

The approach by Solskinnsbakk & Gulla [22] is relying on constructing ontological profiles that contain concept vectors. However, when creating the concept vectors they are depended on a highly relevant document collection. Furthermore, they also need a collection of non-relevant documents in order to construct negative concept vectors. Both vectors are used in query expansion. Testing shows good results for situations where recall is more critical than precision.

2.2 Information needs and search strategies

There are many studies of users' information needs, their search strategies and behaviour (e.g. [1, 11]) resulting in different classification of search strategies. For instance, Guha et al. [9] distinguish two different kinds of search, namely, navigational search and research search. Navigational search is defined as the one where the user provides a phrase or keywords and expects to find them in the documents, i.e. the user is using a search engine to navigate to a particular document. While in the research search the user provides a phrase or keywords that are intended to denote object or phenomena about which the user wants to gather information, i.e. the user is trying to locate a collection of documents which will provide required information [9].

With the emerging Semantic Web there is envisioned a shift in IR from retrieval of appropriate Web pages to answering questions without extraneous information [14]. This, being separate and important areas in information retrieval and knowledge management, requires robust ontology quality, reasoning, and fine-grained annotation of documents. However, precise question answering is the most ambitious information retrieval task but still inevitable and a required feature of Web search. Therefore, we

consider a fact-finding search being able to partially substitute question answering on the Web. For this reason, we adopt a classification of search tasks into the following categories: fact-finding, exploratory, and comprehensive search tasks [1]. In fact-finding, a precise set of results is important, while the amount of retrieved documents is less important. In exploratory search task, the user wants to obtain a general understanding about the search topic, consequently, high precision of the result set is not necessarily the most important thing, nor is high level of recall [1]. Finally, a concern of comprehensive search task is to find as many documents as possible on a given topic, therefore the recall and precision should be as high as possible.

3. ONTOLOGY-DRIVEN SEARCH

In this section, we elaborate on our approach. We start with an introduction to feature vectors then describe the process to construct FVs and finally finish the section by describing how FVs are used in search.

3.1 Introduction to feature vectors

The development of the approach is inspired by a linguistics method for describing the meaning of objects - the semiotic triangle [16]. In our approach, a feature vector "connects" a concept (entity) to a document collection, i.e. a FV is tailored to the specific terminology used in a particular document collection (see Figure 1). FVs are built considering both the semantics encoded in an ontology and the dominant lexical terminology surrounding the entities in a text corpus. Therefore, a FV constitutes a rich representation of the entities and is related to the actual terminology used in the text corpus. For a more formal definition of a FV, the keen reader is referred to [24].

The process of selecting relevant entities and terms (words) into these sets is elaborated in the Section 3.3, but first the overall architecture of the approach is presented in the next subsection.

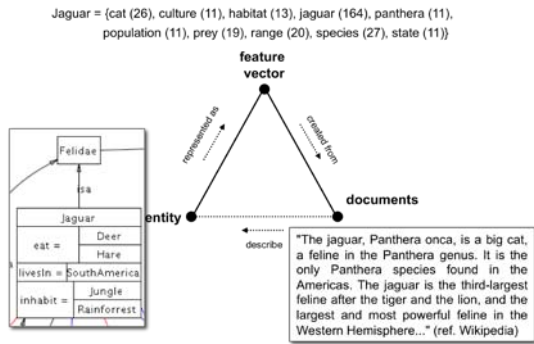


Figure 1. Explanation of a FV by adapted semiotic triangle. In addition, an illustration of a *feature vector* created for the *entity Jaguar* with an ontology fragment (*Animals*²) depicting the *Jaguar entity* together with a text fragment (*documents*) from the Web being related to the entity.

3.2 Architecture

Figure 2 depicts the overall architecture of the ontology-driven information retrieval system. In this section, we will briefly describe the architecture and its components (more details are provided in [25]).

The system consists of both offline and online components (with respect to actual search process). The offline components are used to add and populate new ontologies (Section 3.3) while the online components use the already populated ontologies in search (Section 3.4). The underlying query and indexing system is used both offline and online.

3.3 Feature vector miner

The feature vectors are composed from both the semantics encoded in the ontologies and the surrounding terminology of the entities in a text corpus (the Web). A simplified version of the FV construction process is depicted in Figure 3 (more details can be found in [24]). The process of constructing FVs constitutes main phases.

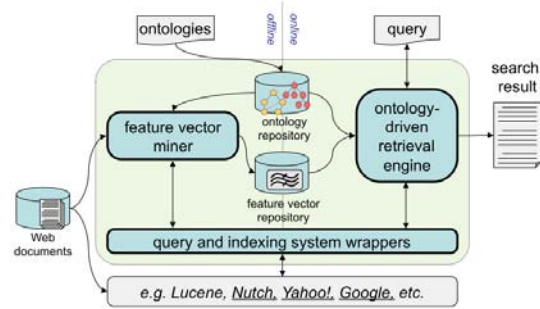


Figure 2. An overview of the ontology-driven information retrieval system and its components.

The main aim of the first phase is to extract and group sets of candidate terms being relevant to each entity. First, an ontology is analysed to find the entities and the relationships among them. Then a query for each entity is composed. The queries are constructed using the entity name and expanded with neighbouring entities (i.e. *parent*, *child*, and/or *other* [24]). The queries are submitted to the underlying search system. The result of this is a set of retrieved documents for each entity. Each document set is clustered to group documents having high similarity. For each cluster a set of candidate key-phrases, noun phrases collocated with the entity, are extracted from the documents of the cluster. These sets candidate key-phrases (represented as a Cluster Feature Vector (CLFV)) associated with each entity are the input to the next and final phase of the process namely identifying and creating the final FVs of the entities.

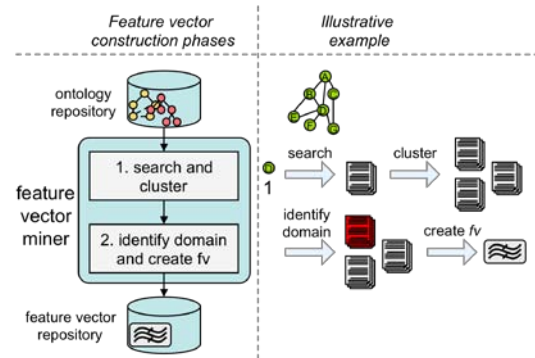


Figure 3. On the left hand side, a simplified version of the Feature Vector Construction process is depicted, while an illustrative example is found on the right hand side.

At this stage of the process, we do not know which of the clusters (CLFVs) for each entity are most relevant to the domain of interest defined by the ontology (e.g. the concept of Jaguar can be part of many different domains like being a car brand, animal, operating system, etc.).

Consequently, the main aim of the last phase is to identify the most relevant clusters w.r.t. the entities defined by the ontology. The hypothesis is that individual clusters having high similarity across ontology entities are with high probability of the same domain (e.g. *Jaguar* w.r.t. *Felidae* depicted in Figure 1). This hypothesis is backed up with observed patterns of collocated terms within the same domain, and consequently different domains will have different collocation pattern of terms. However, the similarity of clusters depends a lot on the quality of the ontology, especially on the semantic distance between the entities. Therefore, the most prominent cluster is found by calculating the similarity between the CLFVs of the current entity with all the CLFVs of the neighbouring entities. Then finally the clusters with the highest score are selected and used to create the FVs for each entity. The result of this process is a FV for each entity with key-phrases that are associated with both the entities and the domain defined by the ontology.

3.4 Ontology-driven retrieval engine

In this section, we will describe the ontology-driven search engine where feature vectors are used to disambiguate search.

First, the user needs to formulate a query. The user can specify one or more entities related to the domain of interest (if no entities are specified then ordinary keyword search is performed). In addition, the user can specify a set of keywords to narrow the search even further (see Figure 6). By differentiating on entities and keywords, the real intention of the user's query can better be interpreted by the underlying machinery and thus present more relevant results.

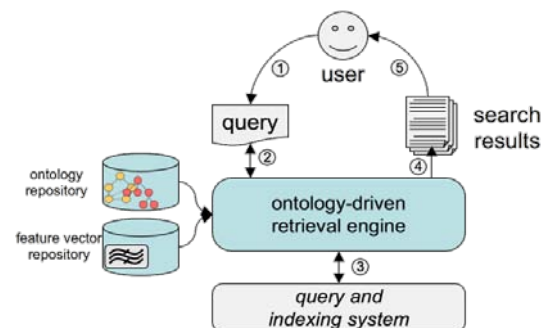


Figure 4. An overview of the search process.

Figure 4 depicts an overview of the different steps of the search process. Firstly, the user initializes a search (1) by submitting a query

$Q = \{e_1, \dots, e_n, k_1, \dots, k_m\}$, where e is an entity part of an ontology O , k is a keyword, n is the total number of entities while m is the total number of keywords, to the *ontology-driven retrieval engine* (2). Then, the retrieval engine identifies the corresponding entities of the ontologies and submits a semantically enriched query $Q' = \{S_{e_1}, \dots, S_{e_n}, k_1, \dots, k_m\}$, where S_e is a semantically enriched (i.e. the entity name and selected neighbouring entities) entity e , to the underlying *query and indexing system* (3). Those query terms with no corresponding entity are treated as ordinary keywords. For each document of the search result, a document feature vector (DFV) is created (3). Then the search result from the underlying search engine is filtered and re-ranked by comparing the similarity between the FV_es of Q and the DFVs. Only those documents having a similarity score with the FV_es of Q above a certain threshold are selected and next ranked according to the similarity scores (4) before presented to the user (5).

4. DESIGN OF EXPERIMENTS

In this section, we present the experiments conducted to evaluate the approach with respect to its sensitivity of both ontology quality and search task.

Table 1. Demographic information about the participants.

Demographic feature	Response	Demographic feature	Response	Demographic feature	Response
Gender	male: 18 (86%)	Amount of keywords in a good query	2 or less: 4 (19%)	Knowledge about ontologies	None: 1 (5%)
	female: 3 (14%)		3: 11 (52%)		Have heard about: 9 (43%)
Age	[18-24]: 13 (62%)	4: 6 (29%)	Have been studying: 5 (24%)		
	[25-29]: 5 (24%)	5: 0 (0%)	Have been using in prototyping: 6 (29%)		
	[30-35]: 2 (9%)	6 or more: 0 (0%)	Practical development: 0 (0%)		
	[40-49]: 1 (5%)				
Web search experience	None: 0 (0%)	Search service preference	Generic Web search: 20 (95%)	Participation in evaluations	First time: 4 (19%)
	Sparse: 0 (0%)		Specialized Web search: 5 (24%)		Sparse: 7 (33%)
	Moderate: 5 (24%)		On-line catalogues: 0 (0%)		Moderate: 8 (38%)
	Extensive as user: 10 (48%)		Specialized digital libraries: 8 (38%)	Extensive as participant: 1 (5%)	
Extensive as user and developer: 6 (28%)	Other (journal site, wikipedia, google specialised search): 3 (14%)	Both as participant & evaluator: 1 (5%)			

4.1 Experiment settings

The participants in our experiment were mainly 4th year students at the Norwegian University of Science and Technology (NTNU) (see Table 1 for demographic information about the participants). 21 subjects participated that were offered payment for used time after full completion of the experiment.

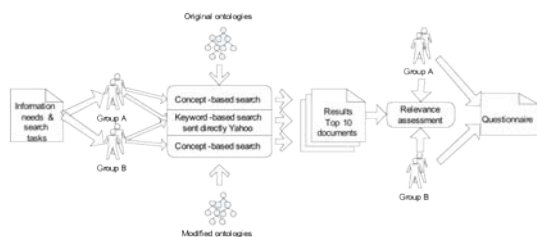


Figure 5. Design of the experiment.

Setting for the experiment is elaborated in Figure 5. The participants of the experiment were given eight topics and descriptions of information needs from four different domains. Queries were specified using keywords and entities from particular domain ontology. They needed to formulate in total 16 queries each (eight submitted to the prototype and eight to the baseline). The system returned 10 top ranked documents for each query, which they assessed based on their relevance to the participants perception of the topic description. After finishing the experiment, the participants completed a questionnaire of 29 questions.

The simulated situation tasks were as follows:

Food & Wine domain

1. *Explorative search task.* What grapes are used to make suitable wines to beef curry¹.
2. *Fact-Finding search task.* Find a perfect dessert wine for a dessert made from chocolate with sweet fruits.

Travel domain

3. *Comprehensive search task.* Try to get an overview of the kind of safaris that are available.
4. *Fact-Finding search task.* Find out the possibilities for a leopard safari.

Animal domain

5. *Explorative search task.* Explore facts about jaguars with the purpose of writing an essay.
6. *Comprehensive search task.* Survey information about jaguars, leopards, and other members of the cat family.

Autos domain

7. *Fact-Finding search task.* Find out about the car brand named Saturn.
8. *Comprehensive search task.* Get an overview of SUVs.

¹ Actually, the users were given a more detailed and verbose description of the topics and information needs in order to define them precisely and avoid ambiguities.

The participants were divided into two groups that used different ontologies² for the same domain (see Figure 5). The first group used the original ontology while the second group used an altered version of the original ontology. The original ontology was altered to include more relations and/or instances to see if this would influence on the search results. All four ontologies were modified by adding instances (all ontologies), specifying additional object properties (travel, animal, and wine ontologies) and refining taxonomical relationships (animal ontology). The results of these changes were different feature vectors generated for the same entities of the two different but still similar ontologies (see Table 2). In summary, group 1 contained 10 participants, while group 2 had 11 participants. In total, the users executed 81 queries using the original ontologies and 92 queries using the modified ontologies, and 152 were simple keyword based queries executed directly to the baseline.

Table 2. Ontology and FV characteristics.

Domain	Ontology version	Ontology characteristics			Feature vectors' characteristics	
		# of concepts	# of instances	# of properties	average length	average cosine similarity
Food & Wine	1	82	155	14	36,66	0,92
	2	83	157	17	38,38	
Travel	1	34	14	10	34,67	0,92
	2	34	29	10	37,26	
Animal	1	51	0	2	33,04	0,78
	2	63	15	8	36,12	
Autos	1	90	321	16	33,27	0,87
	2	91	328	16	33,65	

We choose to use the Yahoo! Web Search API as the backend search engine that consequently also performed as our baseline for our comparison. Ideally, we would use the Text Retrieval Conference (TREC)³ data as baseline. However, we experienced the same problems as d'Aquin et al. [6] in finding good ontologies that covered TREC. In fact, d'Aquin et al. [6] found that those ontologies available on the Web covered only 20 percent of the domains described in TREC (they used the 100 queries from the WT10G test collection). As a result, we choose to use Yahoo! Web Search as baseline and let the participants do a qualitative perceived relevancy of the top 10 results.

We adopted the query scoring and calculation method presented by Brasethvik [3] to measure the qualitative perceived relevancy. The participants needed to mark (as either *trash*,

² The ontologies used are all formalized in OWL and can be found here: <http://research.idi.ntnu.no/IIP/ontologies/>.

³ Text Retrieval Conference (TREC), <http://trec.nist.gov>

irrelevant or duplicate, related, or good) each of top 10 retrieved documents according to perceived relevance. The final relevance score for a query falls into a range [-50, 100] (more details are provided in [23]). The relevance score substitutes a conventional precision metric. We have decided to focus on precision instead of recall since we aimed at improving Web search results, where precision (i.e. relevant documents at top positions) is more important.

4.2 Prototype implementation

A prototype was implemented in Java and the experiments were performed on a standard PC with an Intel™ Pentium processor running Windows™ XP, running Apache Tomcat. A Web user interface similar to a typical search engine found on the Web was created (Figure 6) to make the interface as familiar as possible to the user. To assist the user in finding appropriate entities of the ontology a suggest-like interface was implemented (i.e. when the user started to type a list of suggested entity names were provided that the user could select from).

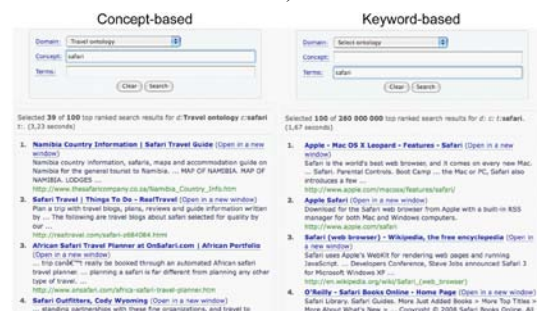


Figure 6. The search user interface of the prototype, concept- search vs. keywords-based.

The implemented prototype was configured to use the Yahoo! Web Search API⁴ as the backend search engine that also performed as the baseline for our evaluation.

4.3 Ontology quality assessment - the EvOQS framework

In this subsection we briefly overview the EvOQS (Evaluation of Ontology Quality for Searching) framework [23] for evaluation of ontology quality in search applications. A part of this framework has been used to assess ontology

quality in the experiment. Here we will focus on that part of the framework, for a complete framework description the keen reader is referred to [23].

The framework defines a stepwise ontology selection procedure and metrics. Ontology quality aspects are defined with respect to the search tasks and search enhancement requirements. The framework adopts earlier discussed (Section 2.2) classification of search tasks into three categories, such as *fact-finding*, *exploratory*, and *comprehensive* search tasks [1]. In this paper, we focus on *search task fitness*. This step concerns evaluation of ontology fitness for a particular search task. For instance, ratio of taxonomic vs. non-taxonomic relationships is important when selecting an appropriate ontology for exploratory and comprehensive search tasks. For instance, in *fact-finding*, a high precision can be achieved by using precise terms or phrases in the query, typically by formulating a query consisting of several terms. In order to enhance results in fact-finding search task provided entities needs to be extended by their instances and datatype properties. Consequently, entities, their instances and properties, are essential here. In *exploratory search*, the user may find topic-related documents by extending simple keyword-based search with parent- and child-entities. In order to cover broader-topics in *comprehensive search*, hypernyms and hyponyms, sibling entities, and semantic relationships are in addition included in the query to cover the most important aspects of the search topic.

The ontology elements that are necessary to support search tasks can be summarized as follows. We compute fact-finding fitness of an ontology as a combined proportion of specified instances and properties vs. specified classes, while explorative fitness measure is based on an average amount of subclasses specified for a class in an ontology or entity cluster. Finally, a metric for comprehensive search fitness is calculated as fraction of object properties, super-classes, subclasses, and sibling-classes w.r.t. the total amount of entities. Recall that all four ontologies were modified by adding instances, specifying additional object properties, and refining taxonomical relationships. The results of these modifications were different FVs created for the equivalent entities.

5. RESULTS ANALYSIS

In this section we analyze the data collected during experiment described in Section 4. We

⁴ Yahoo! Developer Network, <http://developer.yahoo.com>

begin with a generic analysis of the system performance (more information about this generic evaluation can be found in [23]), and then we look at how the modifications of the ontologies changed the feature vectors. Finally, we analyze the sensitivity of the approach to ontology quality and search tasks.

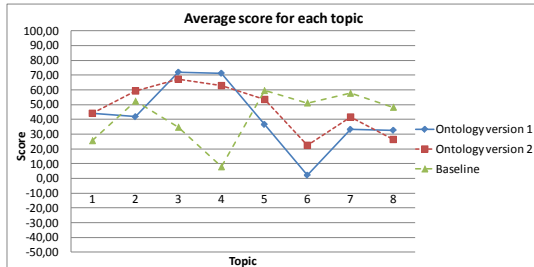


Figure 7. The average scores for each of the eight information-needs described in Section 4.1.

Figure 7 depicts a graph showing how the different ontologies versions influence on the search result relevance score. Recall that ontology version 2 is an altered edition of version 1 having different granularity and level of knowledge specification. The graph shows in general that a more advanced ontology in the sense of having more relations, properties, and individuals does perform better than a similar simpler ontology (see also Table 4).

From Table 3 we can explain the biggest improvement vs. the baseline in topics 3&4 (see Figure 7 and highest scores for ontology usefulness in 3rd column of Table 3). Topic 6 scored lowest on the description and presence of concepts in description (that means it is more difficult to formulate query as consequence), as well topic familiarity and ontology usefulness received third lowest rates – obvious in Figure 7. Furthermore, topic 6 had biggest variance in query length.

Table 3. Mean scores on questionnaire items regarding the experiment. Answers were measured using Likert 5-point scale (from lowest to highest relevance, familiarity, etc.)

Topics	Familiarity w/retrieval tasks	Ontology usefulness	Quality of info needs and task descriptions	Presence of concepts in descriptions
1	2,43	3,48	3,81	2,67
2	2,33	3,43	3,86	2,43
3	2,62	3,57	4,10	2,86
4	2,62	3,76	3,95	2,62
5	2,76	3,38	3,90	2,67
6	2,71	3,14	3,71	2,38
7	2,57	2,81	4,05	2,86
8	2,86	2,95	3,71	2,71

Remark: Lowest values are in bold, while highest in italic.

When observing the length of the queries, it also seems to be a trend that the prototype performs better for shorter queries compared to keyword-based queries which is also observed for other entity-based approaches (e.g. [5]). The entity-based queries were also in general shorter than keyword-based queries.

Another observed pattern is how the users formulate their queries. Recall that the groups were divided into sub-groups. The first group needed to formulate the keyword-based queries prior to the entity-based queries and the other sub-group vice versa. The group formulating the entity-based queries first did in average use 13% less keywords and 14% fewer entities compared to the group formulating the entity-based queries last. Note that a query must contain one or more entities in combination with zero or more keywords to be classified as an entity-based query. However, the group formulating the keyword-based queries first had a tendency to use most of the keywords in the entity-based search as well, consequently having in general longer entity-based queries than the other group. The keyword-based queries for both groups were almost equal in length with a difference of only 2%.

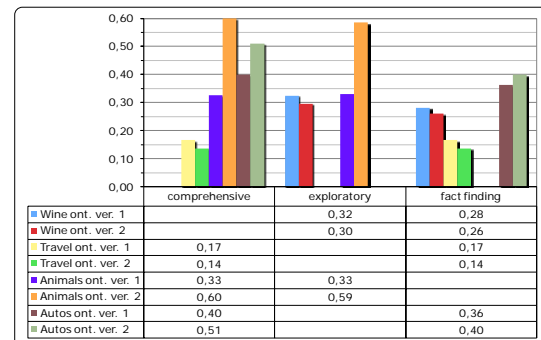


Figure 8. The average neighbouring FV similarity scores of those entities used in the experiment w.r.t. search task.

5.1 Analysis of FV quality and its impact

Figure 8 depicts a graph showing how the average neighbourhood similarity score differs with respect to the search task. In this overview, only those entities that were used in the experiment were considered. Similarity was measured by a standard cosine similarity measure. The graph shows that for the Travel and Wine ontologies the similarity decreases from ontology version 1 to 2 while it is opposite for the Animals and Autos ontologies. High similarity

indicates that an entity's FV is fairly equal to its neighbouring entities' FVs, while a low similarity indicates a more unique FV with respect to its neighbours. We can also observe that the Travel ontology has the lowest neighbourhood similarity scores compared with the other ontologies, but also had the highest relevance scores found in Figure 7 (topic 3 and 4). This indicates that more unique FVs are beneficial versus more general FVs. Therefore, we can assume that the changes done with the Animals ontology, from version 1 to 2, had a negative effect on the uniqueness of its FVs since the neighbourhood similarity increased considerably, but still version 2 performed better than version 1 (see Figure 7 topic 5 and 6).

Table 4. Comparison of mean relevance score of keyword and entity based searches

	Mean relevance score	Diff. from baseline
Keyword-based	42.2	-
Ontology ver. 1	42.1	-0.2%
Ontology ver. 2	46.6	10.5%

From Table 4 we can observe that the modified ontologies significantly increased performance of the prototype. The improvement resulted to be more than 10 percent. Given such significant enhancement we take a closer look at the ontology quality and the role of search tasks in the next subsection.

5.2 Ontology quality impact and search task performance

As a result of the earlier discussed modification of the ontologies, comparing the relevance scores for the original ontologies vs. the modified ones, we found an improvement in mean score that equals to 10.5% (the overall mean relevance for original ontologies score was 42.1 vs. 46.6 for modified ontologies). See Table 4 for comparison of mean relevance scores and Figure 7 for comparison per search topic.

The difference in ontology fitness metrics (see Figure 9) well explains corresponding improvement in performance for the analysed search tasks ($r^2=0.905$). Most significant improvement has been observed in explorative search task, second largest enhancement resulted in fact-finding search task. Addition of more instances and object properties improved the mean relevance score of fact-finding search tasks, while the addition of sub-classes resulted in better performance of exploratory and comprehensive search tasks.

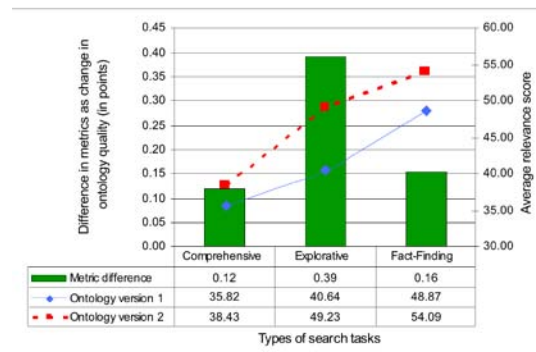


Figure 9. Comparison of ontology quality and search performance based on search tasks⁵.

5.3 Threats to validity

We devote this subsection to discuss possible threats to the results already presented above.

– External validity describes a degree to which the results can be generalized outside the experiment. The experiment has been conducted using only one system (the prototype implementation of the outlined approach), however the conclusions and lessons learnt are applicable to all similar approaches, especially ones using ontology, but limiting its usage to subclass relationships only.

– The case study is executed at the university. However, the experimental tasks and information needs were chosen from the “real world” and most of the test subjects had extensive search experience (16 out of 21 identified having extensive search experience as search users, six of them identified themselves as having a developing experience in addition to extensive search experience). However, we believe that results would be much more in favour of the proposed system if users that are more diverse were selected (subjects were mostly Computer Science students having a bit more sophisticated skills).

– Users provided subjective evaluations. The individuals needed to interpret the experimental materials and tasks according to their experience. The intention was to create an experiment similar to the real usage of Web search, where users formulate and assess relevance by themselves. Experience seemed to be similar for most of the

⁵ Difference of metrics was calculated using equations verbally described in Section 4.3 and formally defined in [23] and metric value of ontology version 1 has been subtracted from the metric value of ontology version 2 (for corresponding search tasks).

individuals. However, we observed a difference (variance) among users' queries and document relevance judgments.

– Fatigue effect. On average 2.5 hours were spent to complete the tasks and fill the questionnaire. Therefore, this effect is not considered relevant.

5.4 Concluding discussion

The participants were in general satisfied with the relevance of the results and the prototype performance. They also found the approach particularly helpful in formulating queries for unfamiliar domains. Analysis of the experiment results shows that users tended to formulate shorter queries for the entity-based approach versus the traditional keyword-based approach. This indicates that they have a prior expectation of such a system compensating the lack of provided information in one way or another.

Furthermore, in the survey, the participants were asked to rate the quality of the results compared to the base system in a scale from 1 (very bad) to 5 (very good), and the mean score was 3.5. This score indicates that the approach for automatic construction of entity feature vectors based on any ontology works quite well and its implementation was not bad either, i.e. the users liked "simplicity" of the ontology-driven search interface.

In summary, we have shown that the proposed approach and its preliminary implementation are apt to improve search performance. However, performance of the approach is dependent on ontology quality (level of knowledge specification). While analyzing the results and trying to find an answer to RQ1, we found difference of 10.5% in improved performance due to enhanced quality of ontology. These findings call for further research on how to tailor FV construction to various search tasks (however, this may require more complicated interface) and research to try different techniques in order to reduce sensitivity of the approach to quality of ontology. Analysis of different search tasks and corresponding performance of the approach on those tasks (RQ2) has shown that certain ontology elements have bigger effect on certain information tasks than other ontology elements (recall Figure 9). Furthermore, the approach has shown the best performance in the fact-finding category of search tasks, having almost 50% higher relevance score if compared to the comprehensive search task (Figure 9). This indicates a need for further research on tailoring

the approach (for instance, tuning FV construction) to the various search task categories and seamless integration with the traditionally simple Web search interface.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an approach that utilizes ontologies to enhance the effectiveness of large-scale search systems for the Web. We have described how such systems can be enriched with adapted ontologies by computing a feature vector for each of the ontology entities that typically includes terms (words) that are associated with the entities. We have briefly described how these FVs are automatically constructed by utilizing the knowledge represented in the ontologies. Finally, we have evaluated the approach.

A prototype was developed and real users evaluated its performance. We used parts of the EvOQS framework [23] to assess ontology fitness and capability to improve ontology-based search. In the framework, evaluation criteria are connected to scenarios of use with a purpose to enhance particular search tasks. We have discussed results of the experiment showing how different ontology quality aspects improve ontology-driven Web search performance. We have found difference of 10.5% in improved performance due to enhanced quality of ontology.

As a future work we will research alternative methods for post-processing of the retrieved documents utilizing the semantic relations in the ontology for better ranking and navigation.

7. ACKNOWLEDGMENTS

This research work is partially funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30. In addition, we would like to thank Jon Atle Gulla (NTNU), Per Gunnar Auran (Yahoo!), and Robert Engels (ESIS) for their support and help.

8. REFERENCES

- [1] Aula, A.: Query formulation in Web information search. In: IADIS International Conference WWW/Internet (ICWI 2003), Algarve, Portugal, 403-410. IADIS (2003)

- [2] Bonino, D., Corno, F., Farinetti, L., Bosca, A.: Ontology driven semantic search. *WSEAS Transaction on Information Science and Application* 1(6), 1597-1605. (2004)
- [3] Brasethvik, T.: Conceptual modelling for domain specific document description and retrieval: An approach to semantic document modelling. PhD thesis, NTNU, Trondheim, Norway (2004)
- [4] Castells, P., Fernandez, M., Vallet, D.: An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering* 19(2), 261-272. (2007)
- [5] Chang, Y., Ounis, I., Kim, M.: Query reformulation using automatically generated query entities from a document space. *Information Processing and Management* 42 (2006) 453-468
- [6] d'Aquin, M., Motta, E., Sabou, M., Angeletou, S., Gridinoc, L., Lopez, V., Guidi, D.: Toward a New Generation of Semantic Web Applications. *IEEE Intelligent Systems* 23 (2008) 20-28
- [7] Esmaili, K.S., Abolhassani, H.: A Categorization Scheme for Semantic Web Search Engines. *Computer Systems and Applications*, 2006. IEEE International Conference on. (2006) 171-178
- [8] Formica, A., Missikoff, M., Pourabbas, E., Taglino, F.: Weighted Ontology for Semantic Search. OTM 2008, Part II, LNCS 5332, Springer-Verlag, (2008) 1289-1303
- [9] Guha, R., McCool, R., Miller, E.: Semantic search. In: Proceedings of WWW2003, ACM Press (2003) 700-709
- [10] Gulla, J.A., Borch, H.O., and Ingvaldsen, J.E.: Ontology Learning for Search Applications OTM 2007, Part I, LNCS 4803, Springer-Verlag (2007) 1050-1062
- [11] Jansen, B., Spink, A., Saracevic, T.: Real life, real users, and real needs: A study and analysis of user queries on the Web. *Information Processing and Management* 36(2) (2000) 207-227
- [12] Jiang, X., Tan, A.-H.: OntoSearch: A Full-Text Search Engine for the Semantic Web. In proc. of the 21st National Conf. on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence (2006)
- [13] Mangold, C.: A survey and classification of semantic search approaches. *Int. J. Metadata, Semantics and Ontologies* 2 (2007) 23-34
- [14] McGuinness, D.: Question answering on the Semantic Web. *IEEE Intelligent Systems* 19(1) (2004) 82-85
- [15] Nagypal, G.: Improving information retrieval effectiveness by using domain knowledge stored in ontologies. In: OTM Workshops. LNCS 3762, Springer-Verlag (2005) 780-789.
- [16] Ogden, C.K., Richards, I.A.: The meaning of meaning: a study of the influence of language upon thought and of the science of symbolism. Kegan Paul, Trench, Trubner & Co, London (1930)
- [17] Panagis, Y., Sakkopoulos, E., Garofalakis, J. and Tsakalidis, A. Optimisation mechanism for Web search results using topic knowledge. *Int. J. Knowledge and Learning* 2(1/2), (2006) 140-153.
- [18] Paralic, J., Kostial, I.: Ontology-based information retrieval. In: Proceedings of the 14th International Conference on Information and Intelligent systems (IIS 2003), Varazdin, Croatia, (2003) 23-28.
- [19] Rose, D., Levinson, D.: Understanding user goals in Web search. In: Proceedings of WWW2004, ACM Press (2004) 13-19
- [20] Scheir, P., Pammer, V., Lindstaedt, S.N.: Information Retrieval on the Semantic Web - Does it exist? : LWA. Martin-Luther-University Halle-Wittenberg (2007)
- [21] Schumacher, K., Sintek, M., and Sauermann, L.: Combining Fact and Document Retrieval with Spreading Activation for Semantic Desktop Search, *ESWC 2008*, LNCS 5021, Springer-Verlag (2008) 569-583.
- [22] Solskinnsbakk, G., Gulla, J.A.: Ontological Profiles in Enterprise Search. *EKAW 2008*, LNAI 5268, Springer-Verlag (2008) 302-317.
- [23] Strasunskas, D., Tomassen, S.L. The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation. *Int. J. Knowledge and Learning*, 4 (4) (2008) 398-414.
- [24] Tomassen, S.L., Strasunskas, D. Construction of Ontology based Semantic-Linguistic Feature Vectors for Searching: the Process and Effect. In *IEEE/WIC/ACM Int. Conf. on Web Int. and Int. Agent Technology (WI-IAT '09)*, IEEE Computer Society, Milano, Italy (2009) 133-138.
- [25] Tomassen, S.L.: Searching with Document Space Adapted Ontologies. In: Lytras, M.D.,

- Carroll, J.M., Damiani, E., Tennyson, R.D. (eds.): *WSKS 2008*, Athens, Greece, September 24-26, 2008. Proceedings, Vol. 5288. Springer-Verlag, Athens, Greece (2008) 513-522
- [26] Wang, H., Zhang, K., Liu, Q., Tran, T., and Yu, Y.: Q2Semantic: A Lightweight Keyword Interface to Semantic Search, *ESWC 2008*, LNCS 5021, Springer-Verlag (2008) 584-598.
- [27] Yang, H-C. A method for automatic construction of learning contents in semantic Web by a text mining approach. *Int. J. Knowledge and Learning* 2(1/2) (2006) 89–105

P7: Cross-Lingual Information Retrieval by Feature Vectors

Publication details

Lilleng, J. & Tomassen, S.L. (2007) Cross-Lingual Information Retrieval by Feature Vectors. In: Kedad, Z., Lammari, N., Metais, E., Meziane, F. & Rezgui Y. (eds.) *Natural Language Processing and Information Systems*, LNCS 4592, Springer, Heidelberg, pp. 229-239.

Cross-Lingual Information Retrieval by Feature Vectors

Jeanine Lilleng and Stein L. Tomassen

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Sem Saelandsvei 7-9, NO-7491 Trondheim, Norway
{jeanine.lilleng, stein.l.tomassen}@idi.ntnu.no

Abstract. This paper investigates query translation in cross-lingual information retrieval, especially the challenges caused by ambiguity and polysemi. We base our ideas on feature vectors and our method uses context during the translation of queries. Achieving good query translation can be difficult, due to short queries lacking context information. We argue that by using information external to the query, like ontologies and document collections, the effect of ambiguity and polysemi can be reduced. Different approaches for translation of these feature vectors are proposed and discussed.

Keywords: cross-lingual information retrieval, query expansion, feature vector

1 Introduction

Cross-lingual information retrieval (CLIR) has been a research area for many years and will be increasingly important. In 2001 Google had more than 2 billion Web pages in their index [1], where approximately half a billion of these was in non-English. In 2005 it was estimated that Google had indexed more than 8.1 billion Web pages [2], while the number of non-English pages was unknown. Additionally, in January 2007 it was assumed that approximately 29% of the Internet users was speaking English [3] compared to while only 17% of the world's population was speaking English. Consequently, when more people start using the Web most of these will be non-English speakers [4]. Considering these figures it will be increasingly important to focus on high-quality CLIR techniques to make the Web truly available for all. In this paper we propose a flexible CLIR approach based on translation of feature vectors (*fvs*).

Monolingual information retrieval, where the language of the query and the document collection are the same, is obviously proven successful since searching is the most used tool on the Web. However, when it comes to cross-lingual information retrieval, where the language of the query and the documents are not necessarily the equal, the situation is quite different. To our knowledge, there are few CLIR systems available for the Web being of satisfactory quality, but for restricted domains (e.g. medicine) CLIR approaches has shown to be more lucrative.

As mentioned, there does exist some CLIR approaches on the Web showing potentials, where probably Babelplex [5] is the most prominent of them. Sadly enough there is little detailed information available for how Babelplex works. Nevertheless, it seems to be using a standard query translation approach where it translates the query terms by using Google Translate [6]. Next, both the original and the translated terms are submitted as two distinct queries to Google and finally the results of each query are presented side by side. However, Babelplex do suffer of the same typical limitations that are common for most CLIR approaches, and that is not being able to disambiguate the terms correctly and hence the translation is often of low quality.

Query interpretation is the first phase of an information retrieval session and the only part of the session that receives clear inputs from the user. Users tend to use very few terms, 3 or less, in their search queries [7, 8]. As a result, the system cannot disambiguate the terms correctly. By adding more relevant terms to the query the domain of interest can to some extent be identified.

However, adding the correct terms is not always trivial, since the user needs knowledge about the terminology used in that particular domain to find those correct terms. Consequently, the users uses few terms that makes it equally difficult for the systems to correctly disambiguate the terms.

For closed or restricted domains CLIR approaches does traditionally produce better result compared to CLIR used in open domains. Typically a domain specific dictionary and thesaurus are used, as a result it is easier for a system to disambiguate the terms of a query and hence produce a better translation. Despite these promising results, they are highly depended on a fairly common terminology being used. Within the oil and gas industry, many companies usually have their own terminology (e.g., all the equipment available). Inconsistent usage of terminology causes problems in documents exchange among the industrial partners. The Integrated Information Platform for reservoir and subsea production systems (IIP) project [9], that partly funds this work, is creating an ontology for all subsea equipment used by the oil and gas industry. A goal of this project is to define an unambiguous terminology of the domain and build an ontology that will ease integration of systems between disciplines.

Ontologies can define concepts and the relationships among them [10] from any domain of interest. Considering multi-disciplinary domains and the big variation of terminology used one of the challenges is adoption of the created ontology to the document space. In our approach [11, 12], we use ontologies to define concepts in a particular domain. We use a query enrichment approach that uses contextually enriched ontologies to bring the queries closer to the user's preferences and the characteristics of the document collection. The idea is to associate every concept of the ontology with a feature vector to tailor these concepts to the specific terminology used in the document collection. Synonyms and conjugations would naturally go into such a vector, but we would also like to include related terms that tend to be used in connection with the concept and to provide a contextual definition of it. Afterward, the *fv*s are used to enrich the query provided by the user.

Since a feature vector includes only those terms found highly related to a concept we believe it can be automatically translated. Based on the semantic relations between the terms in a *fv* it is possible to automatically find a correct translation of each individual term. A correct translation is found and verified by finding an equal semantic relation between the set of translated candidate terms and the original terms of a *fv*. Those candidate terms found to have a similar semantic relation to the original *fv* are selected. The result of this will be a new translated *fv* with equally semantically related terms as the original *fv*.

This paper is organized as follows. In section 2, related work is discussed. In section 3, we describe the proposed approach for translation of feature vectors. Finally, in section 4 we discuss the potentials of this approach and conclude this paper.

2 Related Work

The related work to our approach comes from three main areas. Ontology based IR and cross-lingual information retrieval, in general, and approaches to query expansion, in particular. First, we will present some related work on ontology-based IR and query expansion and then on cross-lingual IR.

Some approaches combine both ontology based IR and the vector space model. For instance, some start with semantic querying using ontology query languages and then use resulting instances to retrieve relevant documents [13]. Nagypal [14] combines ontology usage with vector-space model by extending a non-ontological query. There, ontology is used to disambiguate queries. Paralic et al. [15] describes a similar approach where documents are associated with the concepts in an ontology. The concepts in the query are matched to the

concepts of the ontology in order to retrieve terms and then used for calculation of document similarity.

Most query enrichment approaches are not using ontologies like [16, 17, 18, 19, 20]. Typically, query expansion is done by extending the provided query terms with synonyms or hyponyms (cf. [21]). Some approaches are focusing on using ontologies in the process of enriching queries [22, 23, 24]. However, an ontology in such a case typically serve as a thesaurus containing synonyms and hypernyms/hyponyms, and do not consider the context of each term (i.e. every term is equally weighted).

Qiu et al. [18] is using query expansion based on similarity thesaurus. Weighting of terms is used to reflect the domain knowledge. The query expansion is done by similarity measures. Similarly, Grootjen et al. [17] describes a conceptual query expansion. There, the query concepts are created from a result set. Both approaches show an improvement compared to simple term based queries, especially for short queries.

Adi describes in [20] a commercial search engine that provides three basic search strategies; word, concept, and super-concept search respectively. A concept is represented as a set of words, while a super-concept is a combination of several closely related concepts. The user can mix strategies when searching. Unfortunately, there are not enough details provided by Adi [20] to state how this work.

The approach presented by Ozcan et al. [24] is using ontologies for the representation of concepts. The concepts are extended with similar words using a combination of Latent Semantic Analysis (LSA) and WordNet [25]. The approach gets promising results for short or poorly formulated queries.

Cross-lingual information retrieval is information retrieval with the added challenge of at least two different languages. The early approach to this challenge was to translate the query before the translated query was submitted to the IR system in the same language as the documents to be searched, an example of this is by Quilt [26]. However, ambiguity and polysemy causes significant problems when the query is translated [27]. The challenges are similar to the experienced difficulties in query expansion [28].

Techniques used by Lui et al. [29] to achieve word sense disambiguation in queries might be considered similar to our technique. However, their technique is based on WordNet [25]. This will give good results in general queries, but the WordNet coverage is not very good for more narrow domains (e.g., oil and gas).

3 Approach

In a cross-lingual information retrieval system the query and the documents to be searched are written in different languages. This challenge has one principal solution; translation. The question then becomes what to translate, the query, the documents, or both. Translating the query can be done in runtime, but due to the fact that queries often are very short, it might be difficult to disambiguate the terms. If the documents are translated, more information to disambiguate during the translation is available, but both the required processing time and disk space needed, will be substantial at best. The disk space requirement for n number of supported languages will be n times the original space. The final alternative is to use a common interlingua and translate both the queries and the documents to this language. Obviously this has all the same disadvantages regarding disambiguation as with query translation, but with interlingua only one translation of the queries have to be done and the documents will be independent of the number of languages.

In this paper we will investigate a situation with two languages, and will not investigate an interlingua approach. In addition, we focus on translation done on the query side in order to combine with the existing monolingual IR system. Therefore, this approach will be an extension

of an earlier developed ontology-driven information retrieval (OdIR) system [11, 12] that uses ontologies tailored to the document collection by feature vectors (see Figure 1). The *fv*s are used to enhance the user queries before they are submitted to the IR system.

The expected improvements in query translation caused by the *fv* approach are caused by the information added to the *fv*s from the ontologies and the incorporated document collection (see [11] for further information of the process of creating *fv*s). However, the language resources added to a translation solution are always a limiting factor.

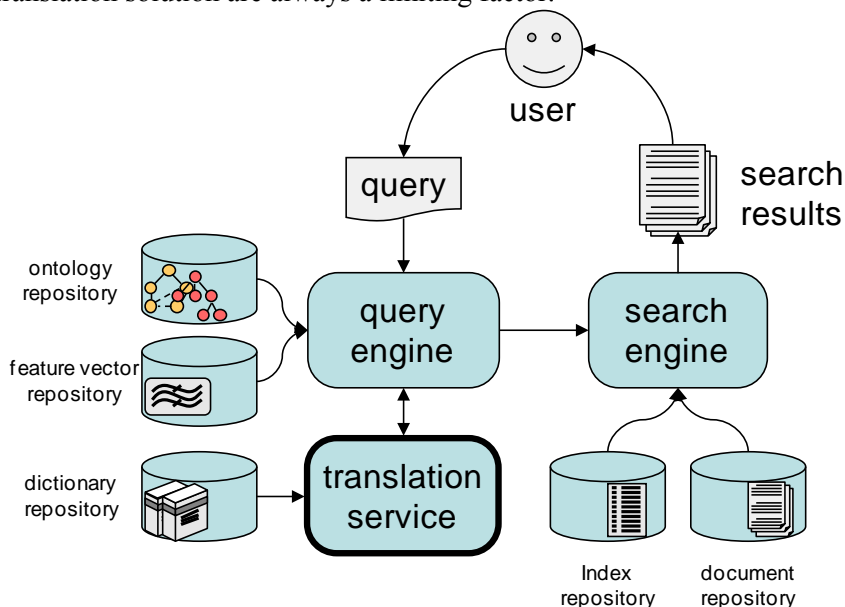


Fig. 1. The overall architecture of the approach. The *translation service* component is an extension to an existing ontology-driven information retrieval system under development, and is the focus of this paper.

3.1 Query translation

Having chosen to translate on the query side reduces the possible solutions somewhat. However, the query passes through three different forms or phases before it is submitted to the IR system; *user query*, *feature vector*, and *enriched query* respectively (see Figure 2). Any of the three forms can be used for translation from the source language to a target language. The chosen phase will affect both the quality of the translation and the number of resources required in the system. Next, the various alternatives will be discussed.

User query. If we choose to translate the user query, a full set of resources is needed for every supported language. This means either a comparable ontology in the target language must be available or a translation of one must be done. Using machine translation will cause reduced quality of both the feature vectors and the final enriched query. One could imagine that translating the ontology and using the target language could create better *fv*s than by translating the *fv*s directly. However, according to Fung [30] semantically similar terms occur in similar context and similar frequency across languages within the same timeframe and domain.

Feature vector. There exists a feature vector for every term in the query. To create these feature vectors both an ontology and the information from statistical analysis of the documents are used. Differences in coverage, granularity, and focus are reduced. Hence, the *fv*s are both domain specific due to the ontology used and adjusted to fit the document collection where the query is to be used. Since the terms of a *fv* are semantically related the possibility for good automatic disambiguation and hence a good translation will be more probable than when translating a few words (e.g. the original query).

Enriched query. The enriched query is a union of all the fvs of all the terms found in the original query. This is the last possible resource for translation. However, since the enriched query is the union of all the fvs , and consequently lacks the distinct fvs used for disambiguation during translation, it is difficult to see how this would be a good alternative.

Based on the pros and cons of the various alternatives discussed above we have chosen to translate form two, feature vectors (see Figure 2). The translation of fvs approach will be discussed next.

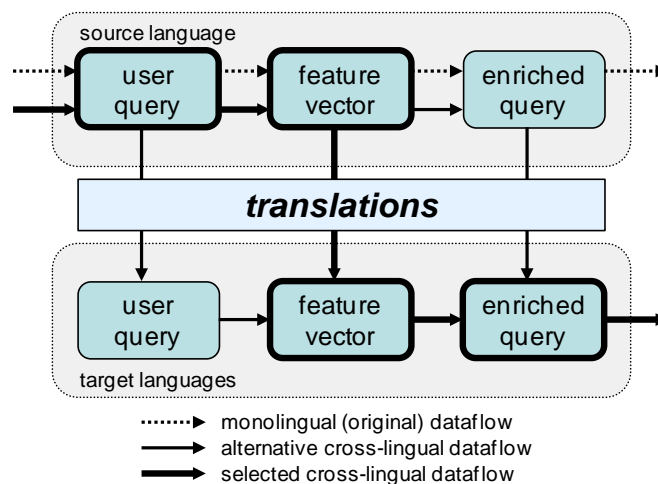


Fig. 2. The various translation approach alternatives. The selected translation approach is shown with bold lines. Note that the original query, depicted as dotted lines, is also sent to the search engine.

3.2 Translation of feature vectors

Before the enriched query can be created, the feature vectors corresponding to the submitted query must be translated to a selected target language. In this section, we will describe two approaches for how these fvs can be translated, but first a method for how we can check for applicability.

A good method to check for applicability seems to first translate the feature vectors to a target language then back to the source language again. If they are equal, it seems reasonable to assume that the translation chosen conserves the semantic content of the feature vectors. Therefore, the hypothesis is that the more equal the content of the translated fvs are with the initial fvs , the more successful is the translation approach.

Figure 3 depicts an explanatory example of an ontology describing trees, some related text fragments from a document collection (Wikipedia [31] is used in this example), and three corresponding examples of feature vectors. These fvs are considered to be of average difficulty, regarding translation. These fvs will also be used to exemplify the translations to German described next.

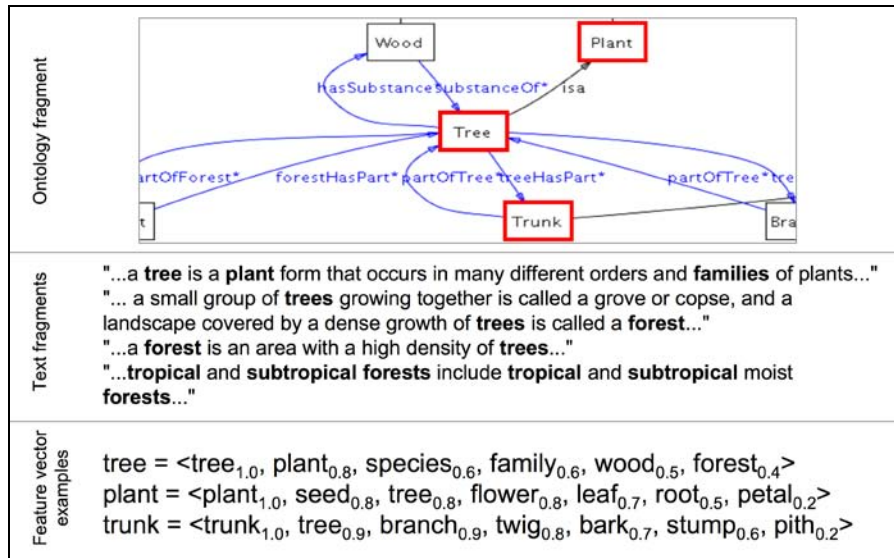


Fig. 3. Explanatory example of three concept feature vectors, including the ontology being used and some related text fragments. These *fvs* are also used to exemplify the translation approaches.

Translation of every term

The intuitive solution is to choose the first suggested translation in a dictionary. This is comparable to submitting one term to a machine translation system and directly use the translated term returned. This approach will provide the translation shown in Figure 4.

plant _(original)	= <plant _{1.0} , seed _{0.8} , tree _{0.8} , flower _{0.8} , leaf _{0.7} , root _{0.5} , petal _{0.2} >
plant _(German)	= <Pflanze _{1.0} , sSen _{0.8} , Baum _{0.8} , Blume _{0.8} , Blatt _{0.7} , Fuss _{0.5} , Blumenblatt _{0.2} >
plant _(English)	= <plant _{1.0} , sow _{0.8} , tree _{0.8} , flower _{0.8} , sheet _{0.7} , feet _{0.5} , petal _{0.2} >
trunk _(original)	= <trunk _{1.0} , tree _{0.9} , branch _{0.9} , twig _{0.8} , bark _{0.7} , stump _{0.6} , pith _{0.2} >
trunk _(German)	= <Kabel _{1.0} , Baum _{0.9} , Zweig _{0.9} , Zweig _{0.8} , Bark _{0.7} , Stummel _{0.6} , Mark _{0.2} >
trunk _(English)	= < cable _{1.0} , tree _{0.9} , arm _{0.9} , arm _{0.8} , barque _{0.7} , snag _{0.6} , pith _{0.2} >

Fig. 4. Translation of every term of the feature vectors, first for `plant` then for `trunk`. For each concept a feature vector being the original (being in English), the German, and finally the one translated back to English again.

If the method retained the semantics of the feature vectors 100%, then the twice-translated *fvs* should be identical to the original *fvs*. Even though both these examples are considered to be of average difficulty only half of the original terms can be found in the twice-translated feature vectors. The results could have been better if the terms found were synonyms with the original words. Unfortunately, in these two examples, they were not. Hence it seems reasonable to conclude that this translation technique is not adequate.

Context dependent translation

Recall that a feature vector is representing a concept and includes only those terms that tend to be used in connection with that concept. We believe the quality of these translated *fvs* can be improved if the semantic information contained in the feature vectors also is used.

Table 1 shows two of the 23 possible direct translations found by the LEO's dictionary [32] for the term `root`. Typically, a term will often have several alternative translations. However, in this example we have selected only two for the term `root`; the one found to be most correct and the one chose by the direct translation approach. For each translation corresponding synonyms are found. The synonyms shown here was found in online dictionaries [33, 34].

Table 1. Two translation matches by the LEO's dictionary for the term `root` and corresponding synonyms for each translation.

Source language	Target language	
Term	Suggested translation	Synonyms
<code>root</code>	fuss	[<i>Fundament, Sockel, Unterbau</i>]
	wurzel	[<i>Wurzelgeflecht, Radix, Wurz</i>] [<i>Anlass, Ansatzpunkt, Ausgangspunkt, Auslöser, Basis, Entstehung, Entstehungsort, Grundlage, Herkunft, Keimzelle, Kristallisationspunkt, Quelle, Ursache, Ursprung, Wiege</i>]

The same process is repeated for all possible translations of the feature vector terms, which gives a large number of alternative final feature vectors. To identify the best translation, the synonym vectors for all the translated terms are compared. Since a lot of additional inaccuracies typically are introduced during translation, we have chosen to do all the comparison in the target language. The combinations of synonym vectors that are most similar are considered correct. Similarity is measured by number of similar words, words that have similar root, or word parts. We expect this to give a better and more context dependent translation.

$\text{plant}_{(\text{original})} = \langle \text{plant}_{1.0}, \text{seed}_{0.8}, \text{tree}_{0.8}, \text{flower}_{0.8}, \text{leaf}_{0.7}, \text{root}_{0.5}, \text{petal}_{0.2} \rangle$
$\text{plant}_{(\text{German})} = \langle \text{Pflanze}_{1.0}, \text{Korn}_{0.8}, \text{Baum}_{0.8}, \text{Blume}_{0.8}, \text{Blatt}_{0.7}, \text{Wurzel}_{0.5}, \text{Blumenblatt}_{0.2} \rangle$
$\text{plant}_{(\text{English})} = \langle \text{plant}_{1.0}, \text{seed}_{0.8}, \text{tree}_{0.8}, \text{flower}_{0.8}, \text{leaf}_{0.7}, \text{root}_{0.5}, \text{petal}_{0.2} \rangle$

Fig. 5. The improved translation approach after including contextual information in the translation process.

The result of this translation approach is shown in Figure 5 for the concept `plant`. Translating `Blatt` back to English can be a challenge, but becomes correct when using the technique described above for the German to English translation as well. In this example the approach retained the semantics of the f_v 100%, that is, the twice-translated f_v was identical to the original f_v . For this reason it seems reasonable to conclude that this translation technique is feasible, but more thorough testing must be done to assess the utility of the approach.

5 Discussion and Conclusion

In this paper we have proposed a novel approach to cross-lingual information retrieval based on feature vectors. We have argued that directly translation of feature vectors can be sufficient for IR applications. However, as the research reported here is still in progress we have not been able to fully implement and evaluate this approach. Even so, we believe the method shows potential because of the quality and the semantic information that these feature vectors possess, which is important and used in the translation process.

To automatically find the correct translation of a term is typically very difficult. The main reason for this is that a term can have many different meanings being highly dependent on the context. Since a typical user tends to use three or less terms in a search query it is difficult, and in most cases impossible, to identify the correct context and hence the correct translation of the query. Consequently, the translation can be totally wrong or all possible translations of the terms must be included. The latter solution will include a lot of noise when searching and is therefore not satisfying. However, for narrow domains the system has some knowledge of the context and consequently the translation can be done more correctly. The terms of a f_v , on the other hand,

are semantically related which provide the system with contextual information that can provide better translation of a query.

The characteristic of a *fv* is dependent on the quality of both the ontology and the document collection being used. However, both the ontology and the document collection are somewhat independent of the approach described in this paper. For instance, there does not exist only one approach for how to create an ontology. One of the reasons for this is that there are many different views of what is considered to be a good ontology. Consequently, the quality of these ontologies will vary a lot depending on the creator. The quality of the documents in a corpus can also vary a lot (e.g., documents found on the Internet). Another important issue is that a good ontology can be applied on a mismatched document collection (e.g., a medical ontology used within the oil and gas domain). All these issues mentioned do have an impact on the final quality of the feature vectors and consequently influence of the translation of these as well, but they are considered all to be external aspects to this approach. In this paper it is assumed that the *fvs* are adequate.

Since we consider the quality of these *fvs* acceptable then we also believe that automatic translation of these can provide satisfying results. Given that a *fv* of a concept only include terms in the document collection that tend to be used in connection with that particular concept, then all those terms are assumed to be semantically related. Based on these semantic relations we believe that it will be possible to find a correct translation of each individual term. To find the likely correct translation of each term we compare with the set of possible translations of the other semantically related terms of the *fv*. Those possible translations that are semantically related are also assumed to be the correctly translated. The result of this will be a new translated *fv* with equally semantically related terms as the original *fv*.

In this paper we have presented two different approaches for how the feature vectors can be translated. The first, *translation of every term*, described a direct translation approach where each term was independently translated of each other. The first translation that the dictionary provided was selected. This approach did not give adequate results, which was not surprising. In the next approach, *context dependent translation*, the semantic relation between the terms was also used in the translation process. In the exemplified result, the twice-translation gave 100% match with the original *fv*. That was only one example and consequently more thorough testing needs to be done before we can conclude how successful this approach is.

As the research reported here is still in progress we need to fully implement the approach for more thorough testing and evaluation. We believe an advantage with this approach is the adaptability to several languages, which can be done by adding other dictionaries and thesauruses. However, that has to be fully tested before we can conclude. We will also have to investigate alternative methods for the translation of the feature vectors. For example, the *context dependent translation* technique described has a major shortcoming; a rather marginal term, with low weighting, has the same influence as more important terms. Therefore, we will investigate methods where the weighting of the terms can be taken into consideration as well.

Acknowledgements. This research work is partly funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR). NFR project number 163457/S30.

References

1. Google: *3 Billion Document Index*: <http://www.google.com/3.html> (22.02.2007)
2. *Search Engine Size Wars V Er upts*: <http://blog.searchenginewatch.com/blog/041111-084221> (22.02.2007)
3. Internet world users by language: <http://www.internetworldstats.com/stats7.htm> (22.02.2007)

4. J. Allan, et al, *Challenges in Information Retrieval and Language Modelling*: report of a workshop held at the centre for intelligent information retrieval 2002
5. Babelplex, <http://babelplex.com/> (22.02.2007)
6. Google Translate, <http://www.google.com/translate> (22.02.2007)
7. Gulla, J.A., Auran, P.G., Risvik, K.M.: *Linguistic Techniques in Large-Scale Search Engines*. Fast Search & Transfer (2002) 15 p.
8. Spink, A., Wolfram, D., Jansen, M.B.J., Saracevic, T.: *Searching the Web: the public and their queries*. J. Am. Soc. Inf. Sci. Technol. 52 (2001) 226-234
9. Gulla, J.A., Tomassen, S.L., Strasunskas, D.: *Semantic Interoperability in the Norwegian Petroleum Industry*. In: Karagiannis, D., Mayer, H.C. (eds.): Proceedings of the 5th International Conference on Information Systems Technology and its Applications (ISTA 2006), Vol. P-84. Köllen Druck+Verlag GmbH, Bonn, Klagenfurt, Austria (2006) 81-94
10. Gruber, T.R.: *A translation approach to portable ontology specifications*. Knowledge Acquisition 5 (1993) 199-220
11. Tomassen, S.L., Strasunskas, D.: *Query Terms Abstraction Layers*. In: Meersman, R., Tari, Z., Herrero, P., al., e. (eds.): OTM Workshops 2006, Vol. 4278. Springer-Verlag, Montpellier, France (2006) 1786-1795
12. Tomassen, S.L., Gulla, J.A., Strasunskas, D.: *Document Space Adapted Ontology: Application in Query Enrichment*. In: Kop, C., Fliedl, G., Mayer, H.C., Métais, E. (eds.): NLDB 2006, Vol. 3999. Springer-Verlag, Klagenfurt, Austria (2006) 46-57
13. Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D.: *Semantic Annotation, Indexing, and Retrieval*. Journal of Web Semantics 2(1), Elsevier, (2005)
14. Nagypal, G.: *Improving Information Retrieval Effectiveness by Using Domain Knowledge Stored in Ontologies*. OTM Workshops 2005, LNCS 3762, Springer-Verlag, (2005) 780-789
15. Paralic, J., Kostial, I.: *Ontology-based Information Retrieval*. Information and Intelligent Systems, Croatia (2003) 23-28
16. Rajapakse, R.K., Denham, M.: *Text retrieval with more realistic concept matching and reinforcement learning*. Information Processing & Management 42 (2006) 1260-1275
17. Grootjen, F.A., van der Weide, T.P.: *Conceptual query expansion*. Data & Knowledge Engineering 56 (2006) 174-193
18. Qiu, Y., Frei, H.-P.: *Concept based query expansion*. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. ACM Press, Pittsburgh, Pennsylvania, USA (1993) 160-169
19. Chang, Y., Ounis, I., Kim, M.: *Query reformulation using automatically generated query concepts from a document space*. Information Processing and Management 42 (2006) 453-468
20. Adi, T., Ewell, O.K., Adi, P.: *High Selectivity and Accuracy with READWARE's Automated System of Knowledge Organization*. Management Information Technologies, Inc. (MITi) (1999)
21. Chenggang, W., Wenpin, J., Qijia, T. et al.: *An information retrieval server based on ontology and multiagent*. Journal of computer research & development 38(6) (2001) 641-647.
22. Ciorăscu, C., Ciorăscu, I., Stoffel, K.: *knOWLer - Ontological Support for Information Retrieval Systems*. In Proceedings of Sigir 2003 Conference, Workshop on Semantic Web, Toronto, Canada (2003)
23. Braga, R.M.M., Werner, C.M.L., Mattoso, M.: *Using Ontologies for Domain Information Retrieval*. Proceedings of the 11th International Workshop on Database and Expert Systems
24. Ozcan, R., Aslangdogan, Y.A.: *Concept Based Information Access Using Ontologies and Latent Semantic Analysis*. Technical Report CSE-2004-8. University of Texas at Arlington (2004) 16
25. WordNet, <http://wordnet.princeton.edu/> (22.02.2007)
26. M. W. Davis and W. C. Ogdan, *QUILT: implementing a large-scale cross-language text retrieval system* 20th annual international ACM SIGIR conference on Research and development in information retrieval, Philadelphia, Pennsylvania, United States, 1997
27. H.-H. Chen, et al, *Resolving translation ambiguity and target polysemy in cross-language information retrieval* Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, 1999
28. C. Stokoe, et al, *Word sense disambiguation in information retrieval revisited* Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, 2003

29. S. Liu, et al, *Word sense disambiguation in queries* Proceedings of the 14th ACM international conference on Information and knowledge management, Bremen, Germany, 2005
30. P. Fung, *A Statistical View on Bilingual Lexicon Extraction: From Parallel Corpora to Non-parallel Corpora*, 1998
31. Wikipedia, <http://en.wikipedia.org> (22.02.2007)
32. Deutch - English Wörterbuch <http://dict.leo.org/> (14.02.2007)
33. Duden <http://www.duden-suche.de/> (14.02.2007)
34. Das digitale Wörterbuch der deutschen Sprache des 20. Jh. <http://www.dwds.de/> (14.02.2007)

P8: Scenario-Driven Information Retrieval: Supporting Rule-Based Monitoring of Subsea Operations

Publication details

Strasunskas, D. and Tomassen, S. L. (2007) Scenario-Driven Information Retrieval: Supporting Rule-Based Monitoring of Subsea Operations, *Information Technology and Control*, 36(1A), pp.87-92.

SCENARIO-DRIVEN INFORMATION RETRIEVAL: SUPPORTING RULE-BASED MONITORING OF SUBSEA OPERATIONS

Darijus Strasunskas and Stein L. Tomassen

*Dept. of Computer and Information Science, Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
{dstrasun, steint}@idi.ntnu.no*

Abstract. The production systems used by the subsea petroleum industry are knowledge and information intensive. Any problem needs to be solved quickly and efficiently avoiding decommissioning or waiting for the symptoms to be escalated. This requires precise information to be supplied on-time. For this reason we have proposed rule-based monitoring of device performance. However, covering all possible cases by rules is a labour-intensive and not trivial task. Therefore, in this paper we propose a scenario-driven information retrieval approach to complement rule-based monitoring. The main objective is to automatically formulate a query that is sent to a vector-space model information retrieval engine every time incomplete inference happens, i.e. when a specific case has no rules defined.

Keywords. Semantic technology, ontology, rule-based inference, information retrieval, integrated operations.

1. Introduction

An industry-driven consortium launched the Integrated Information Platform (IIP) project [4, 11] in 2004. The project's primary objective is to extend and formalize an existing terminology standard for the petroleum industry, ISO 15926 [7]. Using OWL Full sublanguage, this standard is transformed into a real ontology that provides a consistent unambiguous terminology for subsea petroleum production systems. The ontology is, among others, used in monitoring of drilling and production processes.

The production systems used by the subsea petroleum industry are knowledge and information intensive. When a well is put into operation, the production has to be monitored closely to detect any deviation or problems. Any problem needs to be solved quickly and efficiently avoiding decommissioning or waiting for the symptoms to be escalated. Operators' task is actually even more complicated since analysis of a particular problem may involve hundreds of potential causes and require the consultation of a large number of documents.

Therefore, in this paper we propose a scenario-driven information retrieval

approach that complements rule-based condition monitoring of subsea devices. The objective of this paper is to elaborate on task-specific information retrieval and how it can be integrated in rule-based systems in order to support incomplete inference, employing scalability and efficiency of vector space retrieval engines.

The paper is structured as follows. Next we introduce the IIP project. Later we describe a motivating scenario for our approach. Then we elucidate our approach to integration of rule-based notification and task-specific information retrieval. Before concluding the paper, we overview related work.

2. The IIP project

The Integrated Information Platform project is a collaboration project between companies active on the Norwegian Continental Shelf and academic institutions, supported by the Norwegian Research Council. Its long-term target is to increase petroleum production from subsea systems by making high quality real-time information for decision support accessible to onshore operation centres.

The IIP project [4] addresses the need for a common understanding of terms and

structures in the subsea petroleum industry. The objective is to ease the integration of data and processes across phases and disciplines by providing a comprehensive unambiguous and well accepted terminology standard that lends itself to machine-processable interpretation and reasoning. This should reduce risks and costs in petroleum projects and indirectly lead to faster, better, and hence cheaper decisions.

3. Illustrative scenario

Consider a production operator monitoring the production efficiency of a well in the area of oil and gas exploration and production. She is located in a control room with several monitors showing the status of the wells. In such a control room, there are constant alarms of some sort with varying degree of importance. One of the most important responsibilities of the production operator is to look for tendencies among these alarms. One or more of these alarms can indicate an upcoming serious problem that might be handled in advance and hence avoiding a potential disaster. If she can lower the risk of these potential problems by acting quickly to those relevant alarms, the production can continue smoothly. Therefore, retrieval of the right information at the right time is an essential task here.

Continuing the scenario, consider the production engineer noticing a tendency of alarms indicating that the temperature at a choke inlet is increasing. Therefore, she has to find out diagnosis and a solution to this problem. On one of her many displays she sees that one of the alarms is related to the choke that is a part of a “*christmas tree*” installation, i.e. a component found among subsea equipment (see Figure 2c, visualization of the concepts/equipment classes related to “*christmas tree*”). She searches for possible cause and dependent measures in order to find a diagnosis and feasible solution to the problem.

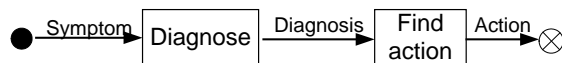


Figure 1. Illustrative activity

Simplified scenario for exemplification of the illustrational case and our approach is denoted in Figure 1. There “*Diagnose*” and “*Find action*” are the main tasks. An actual activity is more complicated [6] involving data pre-processing, mapping to ontology classes, tendency analysis, etc. However, for exemplification purpose we adopt a simplified process containing the most troublesome tasks for full automation.

4. An approach to scenario-driven information monitoring

There are envisioned several application areas of the subsea oil and gas production ontology. Interoperability in the highly multidisciplinary petroleum industry is the main goal [4], while the tasks of ontology-driven information retrieval [15] and rule-based notification [14] have main focus when it comes to supporting routine operations by information retrieval (IR). The rule-based approach is mainly applied to condition monitoring of subsea production. However, not all possible cases can be encoded in rules before hand. Furthermore, here information retrieval should be adjusted to the scenario, since precision of the retrieved information is very important. Therefore, here we present an approach to complement rule-based monitoring with a task-specific and ontology-based information retrieval.

Next we shortly introduce rule-based reasoning for condition monitoring followed by more detailed discussion on task-specific information retrieval. We elaborate on main principles and components of the integrated system.

4.1. Rule-based monitoring

A full case of condition monitoring consists of three main steps [14]: *Data processing*, *Health assessment* and *Treatment planning*. The data processing step takes care of analysis of data streams (Figure 2a, illustrates Daily Product Report - DPR) and mapping the actual measurements to data model (the ontology based on ISO 15926 and

other standards regulating the petroleum domain). The output of this step is a detected state of equipment, for instance, an increased temperature measured at a choke inlet, i.e. identification of symptom.

Having an identified tendency (symptom), next step is *health assessment*, i.e. inference of diagnosis. This step is heavily based on the rules and involves most of reasoning. The rules are used to identify possible causes, infer a diagnosis and finally lead to an action (treatment). At this step we employ rules defined in SWRL (Semantic Web Rule Language) [5]. For instance, *if a choke has a temperature sensor and temperature is equal or above the maximum operating temperature then the choke is in critical state*. This rule is illustrated below using SWRL built-in predicate `swrlb:greaterThanOrEqual` [5], and incoming data in XML format are exemplified in Figure 2a, measure class definition in Figure 2b. Then rule defining dependencies among measurement classes is used to infer diagnosis, as follows.

$$\begin{aligned} &hasTemperatureSensor(?x,?y) \wedge hasTemp(?y,?temp) \wedge \\ &hasMaximumOperatingTemp(?x,?maxtemp) \wedge \\ &swrlb:greaterThanOrEqual(?temp,?maxtemp) \\ &\rightarrow inCriticalState(?x,?temp) \end{aligned}$$

The *treatment planning* step takes care of the last two activities in the condition monitoring cycle, i.e., maintenance planning and actions that need to be taken in order to resolve the situation. This step either notifies the responsible controller who needs to perform actions (e.g. *increase choke opening by 10%*) or executes an action automatically.

```
<witsml:facility>
<witsml:name kind="wellhead" namingSystem="EnergyComponents">...
</witsml:name>
<witsml:facilityParent1 kind="well" namingSystem="EnergyComponent..
</witsml:facilityParent1>
<witsml:facilityParent2 kind="template" namingSystem="EnergyCompo..
</witsml:facilityParent2>
<witsml:unit>ASG-A_L-3H_wellhead</witsml:unit>
<witsml:contextFacility kind="well" namingSystem="EnergyComponent..
</witsml:contextFacility>
<witsml:flow>
<witsml:name>ASG-A_L-3H_wellhead_production</witsml:name>
<witsml:kind>production</witsml:kind>
<witsml:port>L-3H_wellhead_outlet</witsml:port>
<witsml:qualifier>allocated</witsml:qualifier>
<witsml:temp uom="degC">116.95241</witsml:temp>
<witsml:pres uom="bar">147.76852</witsml:pres>
<witsml:portDiff>
<witsml:port>ASG-A_L-3H_portdiff</witsml:port>
<witsml:presDiff uom="bar">45.54977</witsml:presDiff>
<witsml:tempDiff uom="degC">5.83645</witsml:tempDiff>
<witsml:chokeRelative uom="%">67.48616</witsml:chokeRelative>
</witsml:portDiff>
</witsml:flow>
</witsml:facility>
```

Figure 2a. A fragment of Daily Production Report in XML¹

```
<Class ID="ABD134">
<subClassOf resource="&iso15926-4;Choke"/>
<iso15926-4:maximumOperatingTemperature>
<iso31:Temperature>
<iso1000:celsius>
300.0
</iso1000:celsius>
</iso31:Temperature>
</iso15926-4:maximumOperatingTemperature>
etc.
</Class>
```

Figure 2b. Definition of maximum operating temperature for choke

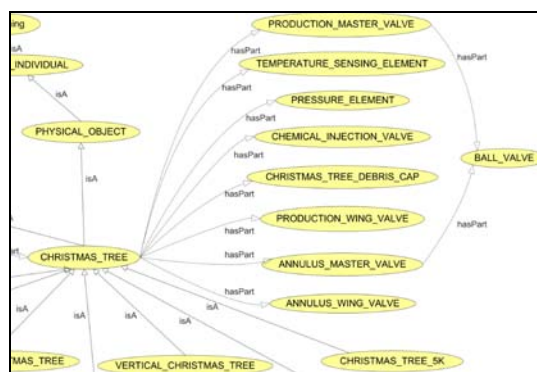


Figure 2c. A fragment of subsea oil and gas production ontology, based on ISO 15926

However, there is a great challenge to completely define rules for all possible dependencies between measures and corresponding actions needed to take to resolve the problematic situations. Operation controller can always refer to manuals or search in a document repository. However, switching between systems or changing the working way requires a considerable amount of time. Therefore, it is a desirable extension

¹ Here, WITSML – Wellsite Information Transfer Standard Markup Language, see <http://www.witsml.org/>

of the current systems to tightly integrate rule-based condition monitoring with information retrieval.

4.2. Scenario-driven information retrieval

In order to complement rule-based monitoring, we propose a scenario-driven information retrieval that is evoked every time incomplete inference happens, i.e. when a specific case has no rules defined. The main objective is to automatically formulate a query that is sent to a vector-space information retrieval engine. Consequently the query should be adjusted to the corresponding tasks. In this subsection, we will first describe the information and knowledge resources that enable us to formulate task-specific queries and then we will present the scenario-driven information retrieval procedure.

For this purpose we adapt our ontology-driven information retrieval [15] method to support rule-based processes of production monitoring. The idea [15] is to construct a *feature vector (FV)* for each of the concepts defined in an ontology. Feature vectors are used to align concepts to the terminology of a document collection and later used for query refinement. This is done by exploiting the ontological structures (i.e. the semantic relationships between concepts) and computing statistical co-occurrence of words that are associated with the concepts in the document collection. These associated terms that often appear together with a particular concept from an ontology constitute the basis for a feature vector. The process of FV construction is elaborated in [16]. However, here we exemplify how a task-specific feature vector is created.

As said, feature vectors provide interpretations of the concepts with respect to the document collection. Synonyms and conjugations would naturally go into such a vector, but also related terms that tend to be used in connection with the concept are included to provide a contextual definition of it. This allows us to tailor the concepts

defined in an ontology to the terms actually used in a document collection.

Having the ISO 15926 standard specified as an ontology, we relate discipline- and task-specific terminology to domain concepts. Each task has a term denoting its scope and, partially, a goal. For instance, the task “*Diagnose*” (Figure 1) has a goal to find a cause and diagnosis for a particular symptom. Therefore, we take this task-specific term (concept), and expand it by adding related terms from the thesaurus for the oil industry. In this case, adding terms and phrases as “reason, problem source, origin of problem, cause, etc.” This set of related terms is used as a main input for computing a task-specific feature vector.

Figure 3 illustrates the main components used in construction of the task-specific feature vectors, while more detailed FV construction process is described in [16]. Here, scenarios and related task-specific terms are extracted from a workflow repository, and expanded by a set of related terms (mainly using synonyms, hypernyms and hyponyms) from oil industry thesaurus. Then, task-specific feature vectors (FV_t) are computed for each pair $\langle c, t \rangle$, where c is a concept name (e.g., from the IIP ontology, see Figure 2c for exemplification of the ontology) and t is a task-specific term. Task-specific feature vectors are built based on statistical co-occurrence of a task-specific terminology together with the concepts from ontology.

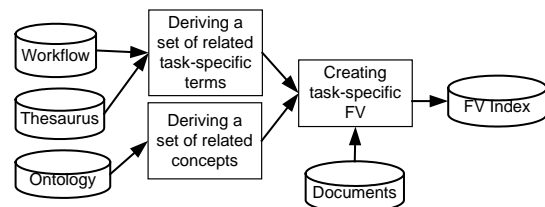


Figure 3. Main components in task-specific feature vector computation

Consider an experience report as follows², where underlined are statistically significant co-occurrence of terms related to “*choke*”, while bold font emphasises the

² Retrieved from Society of Petroleum Engineers, <http://www.spe.org/>.

terms related to the tasks (e.g., action – emergency shutdown, halt; diagnosis – form): “In the summer of 2004, the gas flowline was operated with the subsea choke wide open, controlling the flowline with the topside choke (to control slugging). The manifold pressure was nearly 4,100 psia. In mid-July, an **emergency shutdown (ESD)** was tripped, shutting in the Mica flowlines at the topside boarding valves. [...] methanol injection at the manifold was started and the boarding choke was opened to blowdown the flowline (as per normal startup procedure). Approximately 2 hours after the blowdown was initiated, the subsea choke was opened to start production from the gas well, and as a result, the manifold pressure almost immediately increased 800 psi. In retrospect, this may have been an indication that a hydrate plug had **formed** and that all operations should have been **halted** for further engineering review.”

Then possible task-specific feature vector for a pair <manifold pressure, action> is as follows: {choke, manifold pressure, blowdown, emergency shutdown, ESD, halt, methanol injection}³.

4.3. Interplay between rules and information retrieval

Interaction of the rule-based condition monitoring and notification with ontology-driven information retrieval system is shown in Figure 4. Here searching for relevant information is designed to be supplemental way of interaction with the rule-based system. It is important to enable users to access previous reports and documents related to the problem on-hands. Smooth transition between these two different interaction ways is a challenge as well. Therefore, we propose an automatic query formulation based on either a corresponding inference task that cannot be executed or a returned answer that is incomplete.

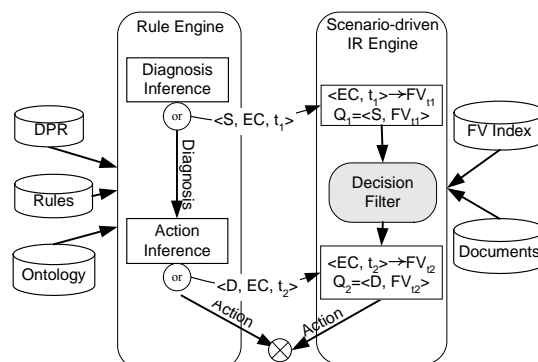


Figure 4. Procedure of interplay between rule and IR engine

A rule engine receives data from Daily Production Reports (DPR), uses rules and ontology to reason about a situation on-hands. If a rule is incompletely defined and no answer can be inferred then the rule engine sends a triple <input, equipment classes, task> to a scenario-driven information retrieval engine. Here an *input* is incoming data to be used in a particular inference task. In Figure 4, “*Diagnosis inference*” uses a set (*S*) of symptoms (e.g. increasing pressure), while “*Action inference*” receives diagnosis (if any) as an input. For instance, after unsuccessful inference of diagnosis, symptoms, related equipment classes (concepts from ontology) and task (task name) are sent to IR engine (see Figure 4).

Then scenario-driven IR engine refines a provided triple and expands the query using corresponding task-specific feature vectors (i.e. FV_{i1} is selected based on the provided concepts (*EC*) and task (t_1)). A component, called *Decision Filter*, has a function to extract a decision from the manually selected document. Actually, the selected relevant document is processed in similar way as it is done while constructing task-specific feature vectors. Just here it is done locally by taking into account only the selected document, i.e. local vs. global document analysis [18]. Here, the first task-specific feature vector (FV_{i1}) is filtered out and reduced to the terms found in the selected document, i.e. $D \subseteq FV_{i1}$.

Query (Q_2) in a second task (finding an action) is formed as Q_1 . First part is a set of diagnosis related terms (*D*) received either from the rule engine (assuming termination

³ Here for simplification purposes, term weight is assumed to be equal.

of reasoning after successful diagnosis inference), or from the previous scenario-driven IR task. Second part is expansion of EC and t_2 by a task-specific feature vector (FV_{t_2}) as in Q_1 .

5. Related work

The problem described here could perhaps be solved using other technologies. For instance, applying fuzzy expert systems and fuzzy reasoning [12] or non-monotonic reasoning [1], that is suitable for reasoning in the cases of incomplete information and knowledge as well inconsistent information.

However, the Norwegian oil industry decided to rely on the Semantic Web technology as a platform for future integrated operations. This comes along with benefits such as semantic interoperability, common inter-disciplinary terminology, etc. Our approach is focused on how to support the underlying information platform.

Liu & Chu [8] have proposed an approach to knowledge-based query expansion to support scenario-specific retrieval of medical documents. Their approach is most similar to ours as they use both statistical co-occurrence and domain knowledge in order to expand the query. However, they rely only on concepts co-occurrence; while we do take into account other terms collocated with a concept of interest. Furthermore, they derive scenario-specific concepts from a knowledge base, namely UMLS⁴ (Unified Medical Language System). They use semantic network to identify scenario-specific concept relations, for instance, having specified that *a medical device and pharmacological substance treat disease*, they are able to identify the semantic type that a concept belongs to and in this way relate concepts “*contact lens*” and “*keratoconus*”⁵ to a scenario that is “*treatment*”. Contrary to them, our approach is based on explicitly defined activities (workflows), where we extract a task-

specific terminology and construct task-specific feature vectors for each concept.

Different approach is chosen by members of the Aksio project [9]. They propose a process driven approach to access experience from daily drilling reports. However, they rely on experts’ annotating the reports and use only ontology concepts and relations between them to expand query. Skalle & Aamodt [13] propose a combined reasoning method (using case-based and model-based reasoning) to support decision in fault diagnosis in oil well drilling.

Furthermore, an important body of work exists in query expansion area (e.g. [2, 10, 17, 18]). Most query enrichment approaches are not using ontologies like [2, 3, 10]. Query expansion is typically done by extending provided query terms with synonyms or hyponyms. Qiu & Frei [10] are using query expansion based on similarity thesaurus. Similarly, Grootjen & van der Weide [3] describes a conceptual query expansion. There, the query concepts are created from a result set. Chang et al. [2] do not use ontologies either but is reliant on query concepts. Two techniques are used to create the feature vectors of the query concepts, i.e. based on document set and result set of a user query [2].

6. Conclusions

The Integrated Information Platform project is one of the first attempts at applying state-of-the-art Semantic Web technologies in an industrial setting. With the ISO 15926 ontology at hand, the industry will have taken the first step towards integrated operations on the Norwegian Continental Shelf. Data can then be related across phases and disciplines, helping people collaborate and reducing costs and risks.

One of the applications developed in IIP is a system for ontology-driven task-specific reasoning and information retrieval. In this paper we presented an approach to task-specific information retrieval to complement rule-based notification. Here, the concepts in the ontology are associated with contextual task terminology in terms of feature vectors

⁴ <http://umlsinfo.nlm.nih.gov/>.

⁵ An eye disease.

tailoring the ontology to the content of the document collection. This adaptation is fundamental in order to provide useful and usable services to a variety of users in the presence of large variations in resources and activities. Further, the feature vector is used to enrich a provided query. Query enrichment by task-specific feature vectors provides means to bridge the gap between query terms and terminology used in a document set, and still employing the knowledge encoded in ontology.

Main advantage of the proposed approach is integration of structured data and knowledge with unstructured information (documents in natural language). However, as future work we will need to experimentally validate our approach in bigger scale. Possible future extensions of the approach would include an experimenting with semantic web services and more tight integration of reasoning and information retrieval. In a current version of the approach there is only one-way communication between rule engine and IR engine. While reasoning on information retrieved from documents could bring additional advantages.

Acknowledgement

This research work is funded by the Integrated Information Platform for reservoir and subsea production systems (IIP) project, which is supported by the Norwegian Research Council (NFR), project number 163457/S30.

References

- [1] **G. Brewka.** *Nonmonotonic reasoning: Logical foundations of commonsense*, Cambridge University Press, Cambridge, 1991.
- [2] **Y. Chang, I. Ounis, M. Kim.** Query reformulation using automatically generated query concepts from a document space. *Information Processing and Management* 42, 2006, 453-468.
- [3] **F.A. Grootjen, T.P. van der Weide.** Conceptual query expansion. *Data & Knowledge Engineering* 56, 2006, 174-193.
- [4] **J.A. Gulla, S.L. Tomassen, D. Strasunskas.** Semantic Interoperability in the Norwegian Petroleum Industry. In D. Karagiannis, H.C. Mayer (eds.), *5th International Conference on Information Systems Technology and its Applications* (ISTA 2006), Vol. P-84. Kollen Druck+Verlag GmbH, Bonn, 81-94.
- [5] **I. Horrocks, P. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, M. Dean.** SWRL: A Semantic Web Rule Language. Combining OWL and RuleML. 2004. Retrieved: March 5, 2007 from <http://www.w3.org/Submission/SWRL/>
- [6] **ISO 13374.** Condition monitoring and diagnostics of machines. Data processing, communication and presentation. 2003. Retrieved: March 5, 2007 from <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=21832>
- [7] **ISO 15926-2.** Integration of life-cycle data for process plants including oil and gas production facilities - Part 2: Data model. 2003. Retrieved: March 5, 2007 from <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=29557>
- [8] **Z. Liu, W.W. Chu.** Knowledge-based query expansion to support scenario-specific retrieval of medical free text. *Information Retrieval* 10(2), 2007, 173-202
- [9] **D. Norheim, R. Fjellheim.** AKSIO - Active Knowledge management in the petroleum industry. *Proceedings of the ESWC 2006 Industry Forum*, Montenegro, 2006.
- [10] **Y. Qiu, H.-P. Frei.** Concept based query expansion. Proceedings of the 16th international ACM SIGIR conference on Research and development in information retrieval. ACM Press, USA, 1993, 160-169
- [11] **N. Sandsmark, S. Mehta.** Integrated Information Platform for Reservoir and Subsea Production Systems. *Proceedings of the 13th Product Data Technology Europe Symposium* (PDT 2004), Stockholm.
- [12] **W. Siler, J.J. Buckley.** Fuzzy Expert Systems and Fuzzy Reasoning. John Wiley & Sons, Inc., 2005.
- [13] **P. Skalle, A. Aamodt.** Knowledge-based decision support in oil well drilling. In: *Intelligent Information Processing II*. IFIP International Conference on Intelligent Information Processing, Beijing, China: Springer Science+Business Media B.V. 2004, 443-455.
- [14] **D. Strasunskas.** Resource Monitoring and Rule-Based Notification. Applications in Subsea Production Systems. In *Proceedings of the 2007 IRMA International Conference*

on Managing Worldwide Operations and Communications with Information Technology (IRMA 2007), Vancouver, Canada, 2007, IDEA Group Publishing.

- [15] **S.L. Tomassen, J.A. Gulla, D. Strasunskas.** Document Space Adapted Ontology: Application in Query Enrichment. In Kop, C. *et al.* (eds.), *11th International Conference on Applications of Natural Language to Information Systems* (NLDB 2006), LNCS 3999, Springer-Verlag, 46-57.
- [16] **S.L. Tomassen, D. Strasunskas.** Query Terms Abstraction Layers. In Meersman, R. *et al.* (eds.): *OTM Workshops 2006*, LNCS 4278. Springer-Verlag, 2006, 1786-1795.
- [17] **E.M. Voorhees.** Query expansion using lexical-semantic relations. *Proceedings of the 17th international ACM SIGIR conference.* 1994, 61-69.
- [18] **J. Xu, W.B. Croft.** Query expansion using local and global document analysis. In *Proceedings of the 19th International ACM SIGIR Conference*, pp. 1996, 4-11.

Appendices

A: Secondary Papers

2010

Conference Papers

Strasunskas, D. & Tomassen, S.L. (2010) On Variety of Semantic Search Systems and Their Evaluation Methods. In: *The Proceedings of the International Conference on Information Management and Evaluation, Academic Conferences Publishing*, pp. 380-387.

2009

Edited Books

Chen, L., Liu, C., Zhang, X., Wang, S., Strasunskas, D., Tomassen, S.L., Rao, J., Li, W.-S., Candan, K.S., Chiu, D.K.W. & Zhuang, Y. (eds.) (2009), *Advances in Web and Network Technologies, and Information Management, APWeb/WAIM 2009 International Workshops: WCMT 2009, RTBI 2009, DBIR-ENQOIR 2009, PAIS 2009, Suzhou, China, April 2-4, 2009, Revised Selected Papers*, LNCS 5731, Springer, Heidelberg.

2008

Journal Papers

Strasunskas, D. & Tomassen, S.L. (2008) The role of ontology in enhancing semantic searches: the EvOQS framework and its initial validation. *Int. J. Knowledge and Learning*, 4(4), pp. 398-414.

Conference Papers

Strasunskas, D. & Tomassen, S.L. (2008) Empirical Insights on a Value of Ontology Quality in Ontology-driven Web Search. In: Meersman, R. & Tari, Z. (eds.) *On the Move to Meaningful Internet Systems: OTM 2008*, LNCS 5332, Springer, Heidelberg, pp. 1319-1337.

Strasunskas, D. & Tomassen, S.L. (2008) On Significance of Ontology Quality in Ontology-driven Web Search. In: Lytras, M.D., Carroll, J.M., Damiani, E. & Tennyson, R.D. (eds.) *Emerging Technologies and Information Systems for the Knowledge Society*, LNCS 5288, Springer, Heidelberg, pp. 469-478.

Tomassen, S.L. (2008) Searching with Document Space Adapted Ontologies. In Lytras, M.D., Carroll, J.M., Damiani, E. & Tennyson, R.D. (eds.) *Emerging Technologies and*

Information Systems for the Knowledge Society, LNCS 5288, Springer, Heidelberg, pp. 513-522.

Posters/Demonstrations

Tomassen, S.L. (2008) Web Search With Document Space Adapted Ontologies. In: Bizer, C. & Josh A. (eds.) *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008)*, 401, CEUR-WS, Karlsruhe, Germany.

2007

Edited Books

Eder, J., Tomassen, S.L., Opdahl, A.L. & Sindre, G. (eds.) (2007) *CAiSE'07 Forum Proceedings*, Tapir Uttrykk, Trondheim, Norway.

Conference Papers

Strasunskas, D. & Tomassen, S.L. (2007) Quality Aspects in Ontology-driven Information Retrieval. In: Khosrow-Pour, M. (ed.) *Proceedings of the 2007 IRMA International Conference on Managing Worldwide Operations and Communications with Information Technology (IRMA 2007)*, IDEA Group Publishing, Vancouver, Canada, pp. 1048-1050.

Workshop Papers

Tomassen, S.L. (2007) Pros & Cons of Applying Industrial Ontologies in Information Retrieval. In Strasunskas, D., Rao, J. & Hakkarainen, S. (eds.) *Proceedings of the 1st International Workshop on Semantic Technology Adoption in Business (STAB'07)*, Tapir Akademisk Forlag, Vienna, Austria, pp. 39-44.

Strasunskas, D. & Tomassen, S.L. (2007) Web Search Tailored Ontology Evaluation Framework. In: *Advances in Web and Network Technologies, and Information Management*, LNCS 4537, Springer, Heidelberg, pp. 372-383.

2006

Conference Papers

Gulla, J.A., Tomassen, S.L. & Strasunskas, D. (2006) Semantic Interoperability in the Norwegian Petroleum Industry. In: Karagiannis, D. & Mayer, H.C. (eds.) *Proceedings of the 5th International Conference on Information Systems Technology and its Applications (ISTA 2006)*, P-84, Köllen Druck+Verlag GmbH, Bonn, Klagenfurt, Austria, pp. 81-94.

Tomassen, S.L., Gulla, J.A. & Strasunskas, D. (2006) Document Space Adapted Ontology: Application in Query Enrichment. In: Kop, C., Fliedl, G., Mayer, H.C. & Metais, E. (eds.) *11th International Conference on Applications of Natural Language to Information Systems (NLDB 2006)*, LNCS 3999, Springer, Heidelberg, pp. 46-57.

Workshop Papers

Tomassen, S.L. & Strasunskas, D. (2006) Query Terms Abstraction Layers. In: Meersman, R., Tari, Z., Herrero P. (eds.) *OTM Workshops 2006*, LNCS 4278, Springer, Heidelberg, pp. 1786-1795.

Tomassen, S.L. (2006) Research on Ontology-Driven Information Retrieval. In: Meersman, R., Tari, Z., Herrero, P. (eds.) *OTM Workshops 2006*, LNCS 4278, Springer, Heidelberg, pp. 1786-1795.

Gulla, J.A., Strasunskas, D. & Tomassen, S.L. (2006) Semantic Interoperability in Multi-Disciplinary Domain. Applications in Petroleum Industry. paper presented to the *The 17th European Conference on Artificial Intelligence (ECAI 2006), Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2006)*, Riva del Garda, Italy, Aug 28th - Sept 1st, 2006.

B: Experiment Invitation Letter

Below is the invitation letter for participation in Experiment I (Section 4.3.1). The letter was sent to a relevant mailing list at NTNU.

Participate in an experiment?

Hi,

would you like to participate in an experiment? The experiment includes evaluating a search engine called WebOdir. WebOdir focus on conceptual search versus more traditional keyword based search.

If you participate in the experiment you will get 150 NOK an hour for your contribution. The experiment will take about 2-3 hours to perform and will be done online from now until the 14th of May. However, but you must have a Norwegian bank account to be paid.

Does this sound interesting? You can find some more information about this evaluation on the following page:

<http://folk.ntnu.no/steint/evaluation/>

You can check out this search engine as well by going to this page:

<http://129.241.110.220>

If you are interested, please send a reply to this email and you will be provided a username and password in addition to some more information on how this evaluation will be conducted.

Kind regards,

Stein L. Tomassen

C: Experiment Introduction Letter

Below is the letter sent to the participants that volunteered to participate in Experiment I (Section 4.3.1). The letter contains an introduction to the experiment.

Evaluation of WebOdIR

Dear evaluator,

Thank you for participating in this experiment and answering the questionnaire. Your feedback is of vital importance for us and is very much appreciated.

Below are instructions how to proceed.

An introduction to this evaluation can be found here:

<http://folk.ntnu.no/steint/evaluation/>

The page above gives a short introduction to the purpose of this evaluation and how it should be performed. Further, it gives a short introduction on how to use WebOdIR and how to conduct the evaluation.

In the experiment you will need to formulate and execute queries on the given topics, then to evaluate retrieved information.

The task description is attached to this email.

In order to login to WebOdIR to do the evaluation you will need a username and password, which you will find below.

Username:

Password:

The post-task questionnaire can be found here:

http://www.surveymonkey.com/s.aspx?sm=1QVEyW8Gom_2fRK7_2fervQLkw_3d_3d

Note, that the first question in the questionnaire is your username (provided above) that you will need to provide.

At last, but not least, after finishing the evaluation, remember to fill out the required information on the list provided by «GreetingLine» to get paid for your work.

Kind regards,

Stein L. Tomassen

D: Introduction to the Prototype

Below is the introduction to the prototype provided to the participants of Experiment I (Section 4.3.1).

WebOdir evaluation information

Introduction

The primary focus of this evaluation is to measure the search quality of WebOdir, a search engine being developed here at IDI NTNU by Stein L. Tomassen. The secondary focus is to evaluate EvOQS, which is a framework to assess fitness of ontologies for use in ontology-based search and is being developed by Darijus Strasunskas and Stein L. Tomassen.

The evaluation has two parts. The first part is about searching. You will be given some information needs were you has to formulate a search query for each information need. In the last part, you will be presented a survey that needs answering. You are free to do some steps of the survey first but the last part of the survey will require that you have done the search part first.

Since some of the questions in the survey are about the quality of the search results, it is therefore recommended to print out that part of the survey (a printable version of those questions can be found here (<http://folk.ntnu.no/steint/evaluation/survey.pdf>)) and make some notes while doing the search part of the evaluation.

Each evaluator will be given an evaluation id. This id is your username when logging into WebOdir and is also asked for in the first question of the survey.

More information about this research?

Some more background information regarding this research can be found here (<http://folk.ntnu.no/steint>) and here (<http://folk.ntnu.no/dstrasun>).

WebOdir user guide

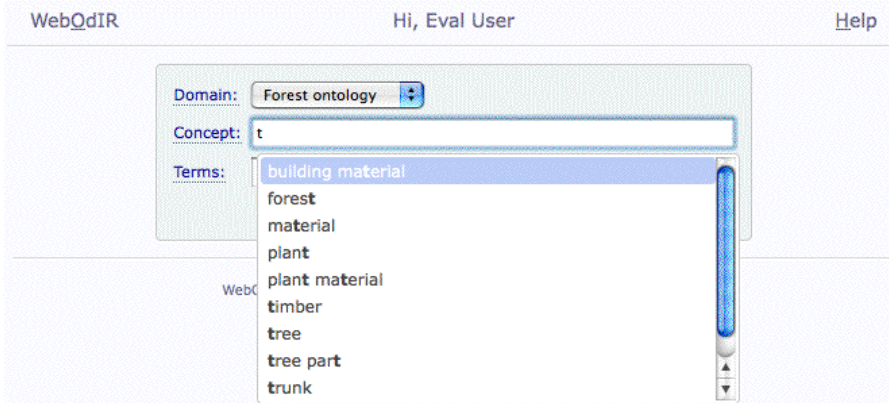
Introduction

The opening screen of WebOdir is shown in the figure below.



You must login to the system to be able to evaluate it, which is done by clicking at the "Sing In" link at the top of the screen. Then you will be presented another window, follow the

instructions. When finally logged in you will see your user name at the top of your screen instead of the "Sing In" link (see the figure below). Now you are ready to evaluate the system.



A search with WebOdIR has three parts that you can specify; these are **domain**, **concepts** and **terms** respectively. With **domain** you can specify the domain of interest, which is specified by a drop down list containing all the domains that are currently available. Each domain has a set of related concepts that are specified in a domain model. A **concept** is focusing on several words that are associated with that concept in contrast to a term that is focus on that single word or term. A concept is specified using a suggest like interface (see the figure above). Either, you can start typing or double clicking the text field, then only relevant concepts for the selected domain are shown. Note, that if you double click the text field and the list of related concepts are shown then only 50 of the concepts are shown at the time. Therefore, if a concept you are looking for is not shown in the list try typing the name of the concept instead. If it is still not shown in the list, then that particular concept is not part of the selected domain model. Only those concepts being part of the list can be used. You can specify more than one concept, which are separated using commas (note, that to use the suggest feature you must add a space after the comma before entering the new concept). In addition, you can specify one or more **terms** which is equal to terms used by e.g. Yahoo!, Google, etc.

Note, to fully utilize the functionality of WebOdIR you must specify both a domain and at least one concept. Terms are not mandatory but can be specified to narrow the search even further if needed. If you do not specify both a domain and at least one concept then an ordinary search (like Yahoo!, Google, etc.) using Yahoo! is performed.

Evaluation in WebOdIR

The figure below shows how the results will look like when logged in as an evaluation user (the user name can be different).

The screenshot shows the WebOdIR interface. At the top, it says "WebOdIR", "Hi, Eval User", and "Help". Below this is a search form with three fields: "Domain:" set to "Forest ontology", "Concept:" set to "tree", and "Terms:" which is empty. There are "Clear" and "Search" buttons. Below the search form is a red header for the "Evaluation" section. Underneath, it says "Selected 10 of 100 top ranked search results for d:Forest ontology c:tree t:. (3,39 seconds)". The results are listed in a table-like format with two columns: the search result and a "Related" dropdown menu. The first result is "1. Care for a Living Christmas Tree (Open in a new window)" with a "Related" dropdown set to "Related". The second result is "2. Project Canopy: Help Trees Help You - Pine Tree State Arboretum (Open in a new window)" also with a "Related" dropdown set to "Related". At the bottom of the results area, there is a red box containing "Select an evaluation topic ID:" followed by a dropdown menu set to "Evaluation topic ID" and a "Submit your evaluation results" button.

At the top of the results area you can see the number of results retrieved. WebOdIR does, in this case, only process the top 100 pages of a search. That is, these 100 pages are evaluated for relevance according to the specified domain of interest, concepts and terms. If only terms are specified then no evaluation is done only presented to the user. Note, that in evaluation mode only the top ten results are shown. Further, you will find the **d**omain, **c**oncepts and **t**erms specified.

On the right hand side of the screen there is a drop down list for each result (see the figure below). You must for each result judge if the result is relevant or not according to your information need specified in the query. The categories of relevance that you can use are:

- **Thrash:** results that have absolutely no relevance at all to the search query.
- **Non-relevant:** results that are considered not being relevant to your query or duplicate of another result.
- **Related:** results that are of the same domain but not exactly what you are looking for.
- **Good:** results that you find useful.

This screenshot shows the bottom part of the search results and the evaluation submission form. The 10th result is "10. Tree Finder and Identification Key - A Quick and Easy Way to Identify ... (Open in a new window)" with a "Related" dropdown set to "Related". Below the results is a red box containing "Select an evaluation topic ID:" followed by a dropdown menu set to "Evaluation topic ID" and a "Submit your evaluation results" button.

When you have considered the relevance for each result, then you must specify the id of the current topic in the drop down list in the bottom of the screen (see the figure above). Finally, you click the submit evaluation to send your results of the evaluation.

This is it, good luck with the evaluation.

Known Issues

WebOdIR is a prototype and have not been extensively tested. Consequently, there will situations where errors will occur. But some hiccups are known:

- Sometimes the user interface looks weird. This is because the style sheet has not been loaded. The reason is unknown but has something to do with the Tomcat server and sessions. This is only a cosmetic bug and not considered serious for this prototype. It can be fixed by clicking the WebOdir link in the upper left corner.
- Sometimes you get an error message saying `"ERROR,OntologyDrivenSearcher::Error calling Yahoo! Search Service: com.yahoo.search.SearchException: Error calling service"` and you get no results. The reason for this error message is usually that a wrong reference to the ontology has occurred. It can be fixed by clicking the WebOdir link in the upper left corner and then try the query again.
- Currently there is a limited number of domain models or ontologies available. The models available are mainly for evaluation and testing purposes only. However, you can add more models if the models are written in OWL (Web Ontology Language), but the you first need access to the Feature Vector Miner of WebOdir. Please contact Stein L. Tomassen for further information.

Resources

- Part of survey for printing (<http://folk.ntnu.no/steint/evaluation/userEvaluation1survey.pdf>)
- WebOdir (<http://folk.ntnu.no/steint/evaluation/userEvaluation1>)

E: Simulated Information Needs

In Experiment I (Section 4.3.1), a set of simulated information needs were created. Below are the descriptions of the information needs provided to the participants.

For each information need, you must first formulate your query using at least one concept. Then you should try to reformulate your query for each information need.

This should be repeated for both the approaches. The first approach you will be using at least one concept from an ontology and alternatively some terms in addition to formulate your query. For the second approach, you will only be using terms that you specify yourself and no concepts found in the ontology will be used.

Food & Wine domain:

1. Imagine that you are going to prepare a dinner for tonight. You plan to make beef curry and would like some wine to drink with this meal. However, you don't know what kind of wine that is suitable. Try to get an overview of what kind of grapes that is suitable.
2. Imagine that you are going to prepare a dessert as well. The main component of this dessert is chocolate but also contains some sweet fruits. You would like to find the perfect dessert wine but don't know which, try to find it.

Travel domain:

3. Imagine that you are going on a vacation and would like to try a safari. You don't know yet which country or what kind of safaris you would like. Try to get an overview of the kind of safaris that are available.
4. Imagine that you like leopards and have decided to go on a leopard safari but don't know where. Explore the possibilities for a leopard safari.

Animal domain:

5. Imagine that you should write an article about jaguars but don't know very much about jaguars. Try to find some facts about jaguars.
6. Imagine that you would also like to write an article about jaguars and leopards and similar kind of cats. Try to get an overview of the cat family.

Autos domain:

7. Imagine that you have heard that the neighbour has bought a new car of the brand Saturn. Further, imagine that you have never heard of this brand before. Try to find some facts about this brand.
8. Imagine that you have become very jealous of your neighbour that recently has bought this beautiful new car. Therefore, you would like to impress your neighbour

as well buy getting a bigger car, a SUV. However, you don't know much about cars; try to get an overview of what SUVs are.

Min 2 queries for each topic and for each approach, which will make the total number of queries to formulate equal to 32.

F: Questionnaire

Holistic Quality Framework

1. Background & descriptive data

This part of the questionnaire is about descriptive data regarding your experience in the areas related to Web search.

*** 1. What is your evaluation id?**

*** 2. What is your gender?**

Female

Male

*** 3. What is your age?**

less than 18

18-24

25-29

30-39

40-49

50 and more

*** 4. How much experience do you have with web search?**

None

Sparse

Moderate

Extensive as user

Extensive as user
and developer

*** 5. Which of the following activities are most relevant for your search activities?**

Select all applicable.

Shopping (information about prices, services, product)

Hobby & interests

Study related (university or self-education)

Work related

Other (please specify)

*** 6. What are your preferred information retrieval services? Specify all applicable.**

Generic Web search engines

Specialized Web search engines

On-line catalogues (categorized information)

Specialized digital libraries

Other (please specify)

Holistic Quality Framework

*** 7. In your opinion, how many terms (in average) must a good query contain in order to expect moderate quality of result? Specify based on your earlier experience.**

- 2 or less
 3
 4
 5
 6 or more

*** 8. Describe your knowledge of ontologies (as a technology)?**

- None
 Have heard about
 Have been studying
 Have been using in prototyping
 Practical development

*** 9. What is your experience of participation in evaluation tasks?**

- This evaluation is the first
 Sparse
 Moderate
 Extensive as participant
 Both as participant and evaluator

2. System Quality

In this and the following parts of the questionnaire you will need to relate your answers to the experiment you have performed.

*** 10. Rank the following features and elements of the evaluated ontology-driven information retrieval system:**

	Very bad	Bad	Fair	Good	Very good
Intuitiveness of interface for query specification	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intuitiveness of results browsing and display	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Easiness of selecting / finding desired concepts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Intuitiveness of concept name meaning	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Easiness of understanding the domain (relations between concepts)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please comment your ranking (optional)

Holistic Quality Framework

*** 11. What of the following possible features of ontology-driven information retrieval do you prefer? Please rank the following features:**

	5th - Least preferable	4th	3rd	2nd	1st - Most preferable	N/A
Visual ontology browsing including all relations between concepts	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Taxonomic ontology browsing including only subclass relations	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Advanced ontology querying using formal ontology query language	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Suggest-as-typed concept selection	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Simple keyword based search	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. Ontology Quality

*** 12. Were all concepts (you needed) present in the ontology?**

- None
 Some
 All

*** 13. Were concept names intuitive?**

- No
 Neither/Nor
 Yes

*** 14. Did you experience any problem due to:**

- Too abstract concepts?
 Too detail concepts?
 Lack of domain knowledge?

4. Query Quality

Provide your answer in a scale of 5.

Holistic Quality Framework					
* 15. Rate your familiarity with the provided information retrieval tasks. Specify for each task.					
	Totally unfamiliar	-	Somewhat familiar	-	Very familiar
Topic 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* 16. How much was ontology usefulness when formulating query for each of the topics?					
	Not at all	...	Neither/nor	...	Very useful
Topic 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* 17. Rate quality of information needs and topics descriptions?					
	Poor	...	Fair	...	Good
Topic 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
* 18. Have all necessary concepts been provided in topics' descriptions given to you?					
	No		Neither/Nor		Yes
Topic 1	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 2	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 3	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 4	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 5	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 6	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 7	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>
Topic 8	<input type="radio"/>		<input type="radio"/>		<input type="radio"/>

Holistic Quality Framework

5. Information Quality

*** 19. Rank five most important criteria for you when you are judging on the Web information relevance and quality.**

	5th - Least important	4th	3rd	2nd	1st - Most important
Visual design and layout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Up-to-date information	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Creditability - provided author's name	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Links to other pages / recommended resources	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Web page title	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Structure of text	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Contents	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Summary	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Presents of readers' comments and rating	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Other (please specify)	<input type="text"/>				

*** 20. Recall the experiment and describe your information satisfaction for each of the topics using scale of 5.**

	Very unsatisfied	Unsatisfied	Neither/nor	Satisfied	Very satisfied
Topic 1	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 2	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 3	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 5	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 6	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 7	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Topic 8	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

*** 21. Rate the following statement with respect how correct they describe your attitude and behaviour:**

	Totally wrong	Wrong	Neither wrong, nor correct	Correct	Very correct
I am willing to spend more time on query specification if results are much better	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I am typically reformulating query several times before I am satisfied with the results	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Holistic Quality Framework

*** 22. Rate the following statements based on your perception.**

	Very bad	Bad	Fair	Good	Very good
Overall prototype performance	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Retrieved information ranking	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Relevance of results	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Quality of results	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. System Satisfaction

*** 23. Describe your general satisfaction with WebOdIR using scale of 5.**

Very unsatisfied
 Unsatisfied
 Somewhat satisfied
 Satisfied
 Very satisfied

*** 24. Compare with other web search engines (e.g. Yahoo!, Google). How would you characterize WebOdIR?**

Significantly worse
 Worse
 Neither worse, nor better
 Better
 Much better

*** 25. What feature of the tested system did you like most?**

- Possibility to specify domain of interest
- Possibility to specify concepts
- Possibility to narrow the search by specifying query terms in addition to concepts
- Simplicity of interface
- Other (please specify)

*** 26. How likely is that you will be using such system in future?**

Very unlikely
 Unlikely
 Difficult to say, depends on specific needs
 Quite likely
 Definitely

*** 27. Rate the following characteristics of the tested system:**

	Much worse	Worse	Neither better, nor worse	Better	Much better
Systems effectiveness (compared with other similar systems)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Systems efficiency (compared with efforts required)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

28. What possible further extensions / features would you recommend to include in the system?

Holistic Quality Framework

29. Please provide your feedback on the evaluation, length and process of experiment, this questionnaire, etc.

7. End of Survey

Thank you very much for your efforts participating in the experiment and filling this questionnaire. Your feedback is very appreciated.

Thanks again!

G: Results of the Questionnaire

Response Summary

Total Started Survey: 29
Total Completed Survey: 28 (96.6%)

Show this Page Only

PAGE: BACKGROUND & DESCRIPTIVE DATA

1. What is your evaluation id?

[Download](#)

	Response Count
Show replies	29
answered question	29
skipped question	0

2. What is your gender?

[Create Chart](#)

[Download](#)

	Response Percent	Response Count
Male <input type="text"/>	86.2%	25
Female <input type="text"/>	13.8%	4
answered question		29
skipped question		0

3. What is your age?

[Create Chart](#)


[Download](#)

	Response Percent	Response Count
less than 18	0.0%	0
18-24 <input type="text"/>	69.0%	20
25-29 <input type="text"/>	20.7%	6
30-39 <input type="text"/>	6.9%	2
40-49 <input type="text"/>	3.4%	1
50 and more	0.0%	0
answered question		29
skipped question		0


4. How much experience do you have with web search? [Create Chart](#) [Download](#)

	Response Percent	Response Count
None	0.0%	0
Sparse	0.0%	0
Moderate <input type="text"/>	17.2%	5
Extensive as user <input type="text"/>	58.6%	17
Extensive as user and developer <input type="text"/>	24.1%	7
answered question		29
skipped question		0

5. Which of the following activities are most relevant for your search activities? Select all applicable. [Create Chart](#) [Download](#)

	Response Percent	Response Count
Shopping (information about prices, services, product) <input type="text"/>	58.6%	17
Hobby & interests <input type="text"/>	93.1%	27
Study related (university or self-education) <input type="text"/>	96.6%	28
Work related <input type="text"/>	48.3%	14
 Show replies Other (please specify) <input type="checkbox"/>	3.4%	1
answered question		29
skipped question		0

6. What are your preferred information retrieval services? Specify all applicable. [Create Chart](#) [Download](#)

	Response Percent	Response Count
Generic Web search engines <input type="text"/>	96.6%	28
Specialized Web search engines <input type="text"/>	20.7%	6
On-line catalogues (categorized information) <input type="checkbox"/>	6.9%	2
Specialized digital libraries <input type="text"/>	37.9%	11
 Show replies Other (please specify) <input type="text"/>	13.8%	4
answered question		29
skipped question		0

7. In your opinion, how many terms (in average) must a good query contain in order to expect moderate quality of result? Specify based on your earlier experience.

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
2 or less <input type="text"/>	20.7%	6
3 <input type="text"/>	48.3%	14
4 <input type="text"/>	27.6%	8
5 <input type="checkbox"/>	3.4%	1
6 or more	0.0%	0
answered question		29
skipped question		0

8. Describe your knowledge of ontologies (as a technology)?

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
None <input type="text"/>	17.2%	5
Have heard about <input type="text"/>	34.5%	10
Have been studying <input type="text"/>	27.6%	8
Have been using in prototyping <input type="text"/>	20.7%	6
Practical development	0.0%	0
answered question		29
skipped question		0

9. What is your experience of participation in evaluation tasks?

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
This evaluation is the first <input type="text"/>	13.8%	4
Sparse <input type="text"/>	34.5%	10
Moderate <input type="text"/>	37.9%	11
Extensive as participant <input type="checkbox"/>	6.9%	2
Both as participant and evaluator <input type="checkbox"/>	6.9%	2
answered question		29
skipped question		0

10. Rank the following features and elements of the evaluated ontology-driven information retrieval system: [Create Chart](#) [Download](#)

	Very bad	Bad	Fair	Good	Very good	Rating Average	Response Count
Intuitiveness of interface for query specification	0.0% (0)	14.3% (4)	21.4% (6)	50.0% (14)	14.3% (4)	3.64	28
Intuitiveness of results browsing and display	0.0% (0)	7.1% (2)	17.9% (5)	46.4% (13)	28.6% (8)	3.96	28
Easiness of selecting / finding desired concepts	0.0% (0)	3.6% (1)	35.7% (10)	39.3% (11)	21.4% (6)	3.79	28
Intuitiveness of concept name meaning	3.6% (1)	0.0% (0)	35.7% (10)	42.9% (12)	17.9% (5)	3.71	28
Easiness of understanding the domain (relations between concepts)	3.6% (1)	7.1% (2)	14.3% (4)	57.1% (16)	17.9% (5)	3.79	28
Show replies Please comment your ranking (optional)							4
answered question							28
skipped question							1

11. What of the following possible features of ontology-driven information retrieval do you prefer? Please rank the following features: [Create Chart](#) [Download](#)

	5th - Least preferable	4th	3rd	2nd	1st - Most preferable	N/A	Rating Average	Response Count
Visual ontology browsing including all relations between concepts	14.3% (4)	17.9% (5)	39.3% (11)	17.9% (5)	7.1% (2)	3.6% (1)	2.85	28
Taxonomic ontology browsing including only subclass relations	11.1% (3)	37.0% (10)	14.8% (4)	14.8% (4)	11.1% (3)	11.1% (3)	2.75	27
Advanced ontology querying using formal ontology query language	57.1% (16)	10.7% (3)	17.9% (5)	7.1% (2)	3.6% (1)	3.6% (1)	1.85	28
Suggest-as-typed concept selection	3.7% (1)	7.4% (2)	11.1% (3)	37.0% (10)	40.7% (11)	0.0% (0)	4.04	27
Simple keyword based search	7.1% (2)	10.7% (3)	10.7% (3)	35.7% (10)	35.7% (10)	0.0% (0)	3.82	28
answered question							28	
skipped question							1	

Show this Page Only

PAGE: ONTOLOGY QUALITY

12. Were all concepts (you needed) present in the ontology? [Create Chart](#) [Download](#)

	Response Percent	Response Count
answered question		28
skipped question		1

12. Were all concepts (you needed) present in the ontology?		Create Chart	Download
None	<input type="checkbox"/>	3.6%	1
Some	<input type="checkbox"/>	75.0%	21
All	<input type="checkbox"/>	21.4%	6
		answered question	28
		skipped question	1

13. Were concept names intuitive?		Create Chart	Download
		Response Percent	Response Count
No	<input type="checkbox"/>	7.1%	2
Neither/Nor	<input type="checkbox"/>	7.1%	2
Yes	<input type="checkbox"/>	85.7%	24
		answered question	28
		skipped question	1

14. Did you experience any problem due to:		Create Chart	Download
		Response Percent	Response Count
Too abstract concepts?	<input type="checkbox"/>	39.3%	11
Too detail concepts?	<input type="checkbox"/>	28.6%	8
Lack of domain knowledge?	<input type="checkbox"/>	64.3%	18
		answered question	28
		skipped question	1

Show this Page Only

PAGE: QUERY QUALITY

15. Rate your familiarity with the provided information retrieval tasks. Specify for each task.		Create Chart	Download		
	Totally unfamiliar	Somewhat familiar	Very familiar	Rating Average	Response Count
Topic 1	25.0% (7)	35.7% (10)	7.1% (2)	2.39	28
Topic 2	25.0% (7)	42.9% (12)	7.1% (2)	2.29	28
Topic 3	7.1% (2)	39.3% (11)	3.6% (1)	2.75	28
		answered question	28		
		skipped question	1		

15. Rate your familiarity with the provided information retrieval tasks. Specify for each task. [Create Chart](#) [Download](#)

Topic 4	7.1% (2)	42.9% (12)	25.0% (7)	21.4% (6)	3.6% (1)	2.71	28	
Topic 5	10.7% (3)	21.4% (6)	32.1% (9)	28.6% (8)	7.1% (2)	3.00	28	
Topic 6	14.3% (4)	25.0% (7)	28.6% (8)	21.4% (6)	10.7% (3)	2.89	28	
Topic 7	14.3% (4)	32.1% (9)	39.3% (11)	7.1% (2)	7.1% (2)	2.61	28	
Topic 8	10.7% (3)	21.4% (6)	39.3% (11)	21.4% (6)	7.1% (2)	2.93	28	
answered question							28	
skipped question							1	

16. How much was ontology usefulness when formulating query for each of the topics? [Create Chart](#) [Download](#)

	Not at all	...	Neither/nor	...	Very useful	Rating Average	Response Count
Topic 1	14.3% (4)	10.7% (3)	21.4% (6)	46.4% (13)	7.1% (2)	3.21	28
Topic 2	10.7% (3)	14.3% (4)	21.4% (6)	42.9% (12)	10.7% (3)	3.29	28
Topic 3	3.6% (1)	14.3% (4)	28.6% (8)	35.7% (10)	17.9% (5)	3.50	28
Topic 4	0.0% (0)	10.7% (3)	28.6% (8)	32.1% (9)	28.6% (8)	3.79	28
Topic 5	7.1% (2)	14.3% (4)	28.6% (8)	39.3% (11)	10.7% (3)	3.32	28
Topic 6	10.7% (3)	21.4% (6)	32.1% (9)	21.4% (6)	14.3% (4)	3.07	28
Topic 7	21.4% (6)	21.4% (6)	21.4% (6)	28.6% (8)	7.1% (2)	2.79	28
Topic 8	0.0% (0)	28.6% (8)	50.0% (14)	17.9% (5)	3.6% (1)	2.96	28
answered question							28
skipped question							1

17. Rate quality of information needs and topics descriptions? [Create Chart](#) [Download](#)

	Poor	...	Fair	...	Good	Rating Average	Response Count
Topic 1	0.0% (0)	7.1% (2)	35.7% (10)	39.3% (11)	17.9% (5)	3.68	28
Topic 2	0.0% (0)	7.1% (2)	35.7% (10)	17.9% (5)	39.3% (11)	3.89	28
answered question							28
skipped question							1

17. Rate quality of information needs and topics descriptions? [Create Chart](#) [Download](#)

	0.0% (0)	7.1% (2)	32.1% (9)	28.6% (8)	32.1% (9)	3.86	28
Topic 3	0.0% (0)	7.1% (2)	32.1% (9)	28.6% (8)	32.1% (9)	3.86	28
Topic 4	0.0% (0)	7.1% (2)	32.1% (9)	35.7% (10)	25.0% (7)	3.79	28
Topic 5	0.0% (0)	7.1% (2)	32.1% (9)	32.1% (9)	28.6% (8)	3.82	28
Topic 6	3.6% (1)	3.6% (1)	39.3% (11)	25.0% (7)	28.6% (8)	3.71	28
Topic 7	0.0% (0)	10.7% (3)	35.7% (10)	17.9% (5)	35.7% (10)	3.79	28
Topic 8	0.0% (0)	10.7% (3)	39.3% (11)	25.0% (7)	25.0% (7)	3.64	28
answered question							28
skipped question							1

18. Have all necessary concepts been provided in topics' descriptions given to you? [Create Chart](#) [Download](#)

	No	Neither/Nor	Yes	Response Count
Topic 1	14.3% (4)	14.3% (4)	71.4% (20)	28
Topic 2	17.9% (5)	10.7% (3)	71.4% (20)	28
Topic 3	7.1% (2)	17.9% (5)	75.0% (21)	28
Topic 4	10.7% (3)	14.3% (4)	75.0% (21)	28
Topic 5	14.3% (4)	14.3% (4)	71.4% (20)	28
Topic 6	21.4% (6)	10.7% (3)	67.9% (19)	28
Topic 7	10.7% (3)	10.7% (3)	78.6% (22)	28
Topic 8	3.6% (1)	21.4% (6)	75.0% (21)	28
answered question				28
skipped question				1

Show this Page Only

PAGE: INFORMATION QUALITY

19. Rank five most important criteria for you when you are judging on the Web information relevance and quality. [Create Chart](#) [Download](#)

	5th - Least important	4th	3rd	2nd	1st - Most important	Rating Average	Response Count
Visual design and layout	31.8% (7)	27.3% (6)	13.6% (3)	18.2% (4)	9.1% (2)	2.45	22
Up-to-date information	12.5% (3)	25.0% (6)	16.7% (4)	25.0% (6)	20.8% (5)	3.17	24
answered question							28
skipped question							1

19. Rank five most important criteria for you when you are judging on the Web information relevance and quality.

[Create Chart](#)
[Download](#)

Creditability - provided author's name	16.7% (2)	8.3% (1)	33.3% (4)	25.0% (3)	16.7% (2)	3.17	12
Links to other pages / recommended resources	0.0% (0)	11.1% (1)	55.6% (5)	33.3% (3)	0.0% (0)	3.22	9
Web page title	16.7% (2)	0.0% (0)	41.7% (5)	16.7% (2)	25.0% (3)	3.33	12
Structure of text	31.3% (5)	31.3% (5)	6.3% (1)	25.0% (4)	6.3% (1)	2.44	16
Contents	13.0% (3)	8.7% (2)	21.7% (5)	13.0% (3)	43.5% (10)	3.65	23
Summary	8.3% (1)	33.3% (4)	0.0% (0)	25.0% (3)	33.3% (4)	3.42	12
Presents of readers' comments and rating	55.6% (5)	22.2% (2)	11.1% (1)	0.0% (0)	11.1% (1)	1.89	9
Hide replies Other (please specify)							1

1. domain name is also important, type errors,

Thu, May 8, 2008 12:43 AM

[Find...](#)

answered question **28**

skipped question **1**

20. Recall the experiment and describe your information satisfaction for each of the topics using scale of 5.

[Create Chart](#)
[Download](#)

	Very unsatisfied	Unsatisfied	Neither/nor	Satisfied	Very satisfied	Rating Average	Response Count
Topic 1	3.6% (1)	10.7% (3)	17.9% (5)	46.4% (13)	21.4% (6)	3.71	28
Topic 2	0.0% (0)	10.7% (3)	28.6% (8)	39.3% (11)	21.4% (6)	3.71	28
Topic 3	3.6% (1)	3.6% (1)	7.1% (2)	46.4% (13)	39.3% (11)	4.14	28
Topic 4	7.1% (2)	10.7% (3)	14.3% (4)	50.0% (14)	17.9% (5)	3.61	28
Topic 5	0.0% (0)	3.6% (1)	21.4% (6)	42.9% (12)	32.1% (9)	4.04	28
Topic 6	10.7% (3)	7.1% (2)	21.4% (6)	28.6% (8)	32.1% (9)	3.64	28
Topic 7	7.1% (2)	14.3% (4)	17.9% (5)	46.4% (13)	14.3% (4)	3.46	28
Topic 8	10.7% (3)	25.0% (7)	21.4% (6)	32.1% (9)	10.7% (3)	3.07	28

answered question **28**

skipped question **1**

21. Rate the following statement with respect how correct they describe your attitude and behaviour:

[Create Chart](#)

[Download](#)

	Totally wrong	Wrong	Neither wrong, nor correct	Correct	Very correct	Rating Average	Response Count
I am willing to spend more time on query specification if results are much better	0.0% (0)	0.0% (0)	21.4% (6)	60.7% (17)	17.9% (5)	3.96	28
I am typically reformulating query several times before I am satisfied with the results	3.6% (1)	10.7% (3)	10.7% (3)	50.0% (14)	25.0% (7)	3.82	28
						answered question	28
						skipped question	1

22. Rate the following statements based on your perception.

[Create Chart](#)

[Download](#)

	Very bad	Bad	Fair	Good	Very good	Rating Average	Response Count
Overall prototype performance	0.0% (0)	0.0% (0)	35.7% (10)	50.0% (14)	14.3% (4)	3.79	28
Retrieved information ranking	0.0% (0)	25.0% (7)	28.6% (8)	46.4% (13)	0.0% (0)	3.21	28
Relevance of results	0.0% (0)	10.7% (3)	46.4% (13)	32.1% (9)	10.7% (3)	3.43	28
Quality of results	3.6% (1)	7.1% (2)	39.3% (11)	46.4% (13)	3.6% (1)	3.39	28
						answered question	28
						skipped question	1

Show this Page Only

PAGE: SYSTEM SATISFACTION

23. Describe your general satisfaction with WebOdir using scale of 5.

[Create Chart](#)

[Download](#)

	Response Percent	Response Count
Very unsatisfied	0.0%	0
Unsatisfied <input type="checkbox"/>	3.6%	1
Somewhat satisfied <input type="checkbox"/>	42.9%	12
Satisfied <input type="checkbox"/>	53.6%	15
Very satisfied	0.0%	0
	answered question	28
	skipped question	1

24. Compare with other web search engines (e.g. Yahoo!, Google). How would you characterize WebOdIR?

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
Significantly worse <input type="checkbox"/>	3.6%	1
Worse <input type="checkbox"/>	21.4%	6
Neither worse, nor better <input type="checkbox"/>	60.7%	17
Better <input type="checkbox"/>	14.3%	4
Much better	0.0%	0
answered question		28
skipped question		1

25. What feature of the tested system did you like most?

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
Possibility to specify domain of interest <input type="checkbox"/>	60.7%	17
Possibility to specify concepts <input type="checkbox"/>	53.6%	15
Possibility to narrow the search by specifying query terms in addition to concepts <input type="checkbox"/>	35.7%	10
Simplicity of interface <input type="checkbox"/>	21.4%	6
 Show replies Other (please specify) <input type="checkbox"/>	7.1%	2
answered question		28
skipped question		1

26. How likely is that you will be using such system in future?

[Create Chart](#)
[Download](#)

	Response Percent	Response Count
Very unlikely <input type="checkbox"/>	3.6%	1
Unlikely	0.0%	0
Difficult to say, depends on specific needs <input type="checkbox"/>	64.3%	18
Quite likely <input type="checkbox"/>	32.1%	9
Definitely	0.0%	0
answered question		28
skipped question		1

27. Rate the following characteristics of the tested system:

[Create Chart](#)[Download](#)


	Much worse	Worse	Neither better, nor worse	Better	Much better	Rating Average	Response Count
Systems effectiveness (compared with other similar systems)	0.0% (0)	17.9% (5)	60.7% (17)	21.4% (6)	0.0% (0)	3.04	28
Systems efficiency (compared with efforts required)	3.6% (1)	14.3% (4)	57.1% (16)	25.0% (7)	0.0% (0)	3.04	28
answered question							28
skipped question							1

28. What possible further extensions / features would you recommend to include in the system?

[Download](#)

		Response Count
Hide replies		18
1. ??	Thu, May 15, 2008 10:54 AM	Find...
2. As said before, better information about the results returned.	Thu, May 15, 2008 10:53 AM	Find...
3. search spesification(filter): scientific, overviews, shopping...	Wed, May 14, 2008 11:08 PM	Find...
4. better interface	Wed, May 14, 2008 4:12 PM	Find...
5. kan bli vrient om man skal lage søkemotoren stor, med kategorier for alt, kan fort bli mer forvirrende enn det blir lettvindt	Wed, May 14, 2008 12:26 AM	Find...
6. Beter UI. The relevance dropdown menu should probably have been a set of checkboxes instead. Not sure if this is applicable for the search engine itself or just for the evaluation.	Tue, May 13, 2008 11:12 PM	Find...
7. Possibly some sort of browser to pick concepts rather than a ajax-search-thingy	Tue, May 13, 2008 7:33 PM	Find...
8. Add the ontology option after a normal search has been performed. Maybe it is easier to decide after the search has been done whether ontologies could be used.	Tue, May 13, 2008 3:59 PM	Find...
9. The possibility to narrow down a search by being presented with related concepts to the one you've already typed in.	Tue, May 13, 2008 2:09 AM	Find...
10. Possibility to combine domains	Mon, May 12, 2008 9:19 PM	Find...
11. Simplify the system. Words like domain, ontology, etc can be confusing for the normal user	Mon, May 12, 2008 5:27 PM	Find...
12. en egen seksjon fo kjøp og pris, se f.eks mine treff på suv-søket.	Sun, May 11, 2008 11:37 PM	Find...
13. I really dont know	Sat, May 10, 2008 12:26 PM	Find...
14. I don't know. I think Google will be better anyway.	Fri, May 9, 2008 9:37 PM	Find...
15. I like Google because it is simple, this system had too many option fields. But, I would probably prefer this one, if I was searching for relevant information for my studies	Thu, May 8, 2008 12:50 AM	Find...
16. browsing or some visualisation of ontology, in order for the user to familiarise with the domain	Wed, May 7, 2008 10:40 PM	Find...
17. More concepts	Wed, May 7, 2008 1:08 PM	Find...
18. Exclusion of concepts, e.g. safari NOT(mac)	Tue, May 6, 2008 5:42 PM	Find...
25 responses per page		
answered question		18
skipped question		11

29. Please provide your feedback on the evaluation, length and process of experiment, this questionnaire, etc. [Download](#)

		Response Count
 Hide replies		27
1.	very easy GUI	Thu, May 15, 2008 10:54 AM Find...
2.	I would just like to say that I think a system like this would probably be very useful for a limited domain (e.g. a computer science domain with subdomains like java, databases etc.), but I'm having a hard time seeing how it will be possible to use a system like this in search engines that has information for a possible extreme amount of domains (e.g. Google).	Thu, May 15, 2008 10:53 AM Find...
3.	It takes more than 5 hours, because I tried to give quality feedback	Wed, May 14, 2008 11:08 PM Find...
4.	It was a god evaluation. It wasent that long. I just hope I helped.	Wed, May 14, 2008 9:10 PM Find...
5.	The process of the experiment was good, and although the length of it was long it was also expected. I think however that all the information needed to get started on the experiment could have been gathered in one place. It was some time consuming when I had to check to emails, with different links in each of them.	Wed, May 14, 2008 7:32 PM Find...
6.	i really liked the fact that i can specify concept and keyterms, but this should be more visible, more feedback that spesifies that you can use the concept, maybe suggested concepts	Wed, May 14, 2008 4:12 PM Find...
7.	spm 27? betyr ikke effectiveness og efficiency det samme? virka bra den her. søker man på f.eks safari på google dukker jo kun mac os opp osv. det vin søket til mat var noe jeg faktisk kan finne på å bruke.	Wed, May 14, 2008 12:26 AM Find...
8.	A bit long	Wed, May 14, 2008 12:01 AM Find...
9.	Lenght was fine. The questionnaire was fine. I could not find a set of "search tassks" assigned to me (maybe i missed an email or something) so i just did a few searches based on my own interest.	Tue, May 13, 2008 11:12 PM Find...
10.	Some of the tasks where more specific than others, don't know if that was intentional or not. The specific ones where easy to just use one term from the task directly whilst others made you think. Length and so on, okay.	Tue, May 13, 2008 7:33 PM Find...
11.	Somewhat difficult to get an overview of what the evaluation would contain. Instead of long emails and webpage with detailed description, a simpler step-by-step guide or todo overview would probably made it easier :)	Tue, May 13, 2008 3:59 PM Find...
12.	No complaints, took a bit of time reading and understanding the system, apart from that, all is good!	Tue, May 13, 2008 2:09 AM Find...
13.	a bit long	Mon, May 12, 2008 9:19 PM Find...
14.	ok evaluation, some questions unclear in questionnaire. spent about two hours	Mon, May 12, 2008 7:42 PM Find...
15.	a very fine combination	Mon, May 12, 2008 6:37 PM Find...
16.	The evaluation was good, both in length and content	Mon, May 12, 2008 5:27 PM Find...
17.	Jeg forsto ikke spm 17 og har derfor bare svart Fair på alle. Er umulig å ikke velge et alternativ i spm 19, jeg valgte et kun for å gå videre. Tok like lang tid som sagt i beskrivelsen (max3t), legge til spm og fornøydhed med resultatene i printversjonen, husker såvidt forskjellen mellom de to søkene mhp resultater. Noen døde linker, disse er merket trash.	Sun, May 11, 2008 11:37 PM Find...
18.	the experiment was a little bit too long. i think you wil get more accurate result if you only gave each person 4 questions	Sat, May 10, 2008 12:26 PM Find...
19.	The evaluation was very bad. The questions are difficult to understand and often use unfamiliar words. It was very hard to figure out how I was suppose to use the search engine for the evaluation. The person making this survey did a terrible job.	Fri, May 9, 2008 9:37 PM Find...
20.	I used about 2.5 hours. The experiment required some preparation (reading, printing out assignment and form), in addition to the survey.	Thu, May 8, 2008 12:50 AM Find...
21.	I think it looks very good. Some questions could give more alternatives or have a different wording. F.ex. the neiter/nor sometimes sounds negative, while the intention is simply stating that it is difficult to specify.	Wed, May 7, 2008 10:40 PM Find...
50 responses per page		
answered question		27
skipped question		2

29. Please provide your feedback on the evaluation, length and process of experiment, this questionnaire, etc. [Download](#)

22. Im afraid i ranked all duplicate links as trash.	Wed, May 7, 2008 1:51 PM	Find...
23. Easy and understandable.	Wed, May 7, 2008 1:08 PM	Find...
24. good	Wed, May 7, 2008 1:07 PM	Find...
25. OK	Tue, May 6, 2008 5:42 PM	Find...
26. Passe lang, kanskje litt vanskelig å tolke enkelte oppgaver	Tue, May 6, 2008 4:54 PM	Find...
27. Eg brukte ca to timar og tjue minuttar på forsøket. Dei siste to query topicane var litt vanskeleg å få gode treff på.	Tue, May 6, 2008 10:00 AM	Find...

50 responses per page

answered question	27
skipped question	2

H: Workshop

First International Workshop on Aspects in Evaluating Holistic Quality of Ontology-driven Information Retrieval *** ENQOIR 2009 ***

The Joint International Conferences on Asia-Pacific Web Conference & Web-Age Information Management
(APWeb-WAIM 2009)

April 1-4, 2009 | Suzhou, China

The ENQOIR workshop targets to deeper understanding and disseminate knowledge on advances in evaluation and application of ontology-based information retrieval (ObIR). The main areas of the workshop is an overlap between three evaluation aspects in ObIR, namely, evaluation of information retrieval, evaluation of ontology quality's impact on ObIR results, and evaluation of user interaction complexity. The main objective is to contribute to optimization of ObIR by systemizing existing body of knowledge on ObIR and defining a set of metrics for evaluation of ontology-based search. The long-term goal of the workshop is to establish a forum to analyze and proceed towards a holistic evaluation method for evaluation of ontology-driven information retrieval systems.

CALL FOR PAPERS

In the recent years, a significant research effort has been devoted to ontology-driven information retrieval. The progress and results in this area offer a promising prospect to improve performance of current information retrieval systems. However, semantics-driven systems are not mainstream-adopted by industry, because there is a lack of adequate evaluation to demonstrate that the benefits of the new technology will overwhelm the payout.

Existing sparse evaluations of ontology-based information retrieval (ObIR) tools report improvement compared to traditional IR systems. However, the results lack indications whether this improvement is optimal. Furthermore, additional sophistication of the ObIR tools adds complexity on user interaction to reach improved results. Consequently, standard IR metrics as recall and precision do not suffice alone to measure user satisfaction because of complexity and effort needed to use the ObIR systems.

Furthermore, evaluation methods based on recall and precision do not indicate the causes for variation in different retrieval results. In addition, there are many other factors that influence the performance of ontology-driven information retrieval, such as ontology quality, complexity of user interaction, difficulty of a searching topic with respect to retrieval, indexing, searching, and ranking methods. The detail analysis on how these factors and their interactions affect a retrieval process can help to dramatically improve retrieval methods or processes.

From other hand, ontology's ability to capture the content of the universe of discourse at the appropriate level of granularity and precision and offer the application understandable correct information is important. An important body of work already exists in ontology quality assessment field. However, most of ontology evaluation methods are generic quality evaluation frameworks, which do not take into account application of ontology. Therefore there is a need for task- and scenario-based quality assessment methods that, in this particular case, would target and optimize ontology quality for use in information retrieval systems.

The purpose of this workshop is to discuss and agree on a set of metrics and hereby lay the foundation for the holistic quality evaluation of ontology-driven information retrieval. Particularly, we strongly encourage submissions dealing with ontology quality aspects and their impact on IR results, evaluation of usability of the ObIR systems, analysis of user behaviour, new evaluation methods enabling thorough and fine-grained analysis of ObIR performance, etc.

TOPICS

All submissions that focus on different aspects of a holistic evaluation of the ontology-driven information retrieval are invited. The main topics of interest are as follows:

- Evaluation of Ontology-driven Information Retrieval
 - Information retrieval evaluation
 - Assessment of annotation quality/labour-load
 - Evaluation and benchmarking techniques and datasets

- Quantitative / qualitative evaluation methods
- Cost/ utility ratio
- Ontology quality aspects in Information Retrieval
 - Ontology quality evaluation
 - Ontology utility
 - Ontology maintenance
 - Quantitative / qualitative evaluation methods
- User acceptance of semantic technology
 - Usability evaluation
 - Quantitative / qualitative evaluation methods
 - Evaluation of human-computer interaction.

SUBMISSIONS & PUBLICATION

We invite submissions of two types: regular papers, and research in progress papers. Papers are restricted to a maximum length of 12 pages (including figures, references and appendices). Submissions must conform to Springer's LNCS format. All accepted papers will be published in a combined APWeb-WAIM'09 workshops volume (as post-proceedings) of Lecture Notes in Computer Science series by Springer.

The extended best papers will be considered for publication in a standard issue of ACM JDIQ (ISSN: 1936-1955). While extended versions of other accepted papers will be considered for publication in a special issue on *Evaluation Aspects of Semantic Search Applications* of the *International Journal on Metadata, Semantics and Ontologies* (ISSN: 1744-2621).

ORGANIZING COMMITTEE

- Darijus Strassunskas (NTNU, Norway)
- Stein L. Tomassen (NTNU, Norway)
- Jinghai Rao (AOL, China)

PROGRAM COMMITTEE

- Per Gunnar Auran (Yahoo! Technologies, Norway)
- Xi Bai (Univ. of Edinburgh, UK)
- Robert Engels (ESIS, Norway)
- Avigdor Gal (Technion, Israel)
- Jon Atle Gulla (NTNU, Norway)
- Sari E. Hakkarainen (Finland)
- Monika Lanzemberger (Vienna Univ. of Technology, Austria)
- Kin Fun Li (University of Victoria, Canada)
- Federica Mandreoli (Univ. of Modena e Reggio Emilia, Italy)
- James C. Mayfield (John Hopkins University, USA)
- Gabor Nagypál (disy Informationssysteme GmbH, Germany)
- David Norheim (Computas, Norway)
- Jaana Kekäläinen (Univ. of Tampere, Finland)
- Iadh Ounis (Univ. of Glasgow, UK)
- Marta Sabou (The Open University, UK)
- Tetsuya Sakai (NewsWatch, Inc., Japan)
- Amanda Spink (Queensland Univ. of Technology, Australia)
- Peter Spyns (Vrije Universiteit Brussel, Belgium)
- Heiko Stoermer (University of Trento, Italy)
- Victoria Uren (The Open University, UK)

DATES

January 11, 2009	Submission of papers
February 2, 2009	Notification about decision
February 20, 2009	Camera-ready versions due

{ <http://events.idi.ntnu.no/enqoir09/>
[enqoir09\[at\]gmail.com](mailto:enqoir09[at]gmail.com) }

I: Ontologies

Ontologies of different granularity were used to measure their effect on the algorithm. All the ontologies are formalised in OWL and can be found at: <http://research.idi.ntnu.no/IIP/ontologies/>. A short description of the ontologies is provided next:

- The Animals ontology is a small ontology that classifies some species, does not contain any individuals, and only has hierarchical properties. The ontology was selected to see the effect of applying the approach on a typical *taxonomy*.
- The Travel ontology is more advanced compared to the Animals ontology by containing individuals and some object properties. As a result, more relationships among the entities are available. The ontology is classified in this work as a *lightweight* ontology.
- The Autos ontology is more advanced than the Travel ontology with more classes, individuals, and object properties. This ontology also uses data properties in contrast to the other ontologies used in this work.
- The Wine ontology is more advanced than the Travel ontology with more individuals and relations. This ontology was originally constructed to test reasoning capabilities. Perhaps, as a result, the ontology contains some entity labels that are not typically found elsewhere (e.g. the entity "McGuinnesso" is according to the ontology, a winery; however a search using Google[®] provides no results of such a winery). Consequently, several entities will not be populated with this ontology. The ontology is, in this work, classified as *advanced* and can, to some extent, indicate the robustness of this approach.