NKISI-ORJI, I. 2019. *Ontology driven information retrieval*. Robert Gordon University [online], PhD thesis. Available from: https://openair.rgu.ac.uk

Ontology driven information retrieval.

NKISI-ORJI, I.

2019

The author of this thesis retains the right to be identified as such on any occasion in which content from this thesis is referenced or re-used. The licence under which this thesis is distributed applies to the text and any original images only – re-use of any third-party content must still be cleared with the original copyright holder.

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

CC BY NC

Ontology Driven Information Retrieval

Ikechukwu Nkisi-Orji

A thesis submitted in partial fulfilment
of the requirements of
Robert Gordon University
for the degree of Doctor of Philosophy

May 2019

Abstract

Ontology-driven information retrieval deals with the use of entities specified in domain ontologies to enhance search and browse. The entities or concepts of lightweight ontological resources are traditionally used to index resources in specialised domains. Indexing with concepts is often achieved manually and reusing them to enhance search remains a challenge. Other challenges range from the difficulty in merging multiple ontologies for use in retrieval to the problem of integrating concept-based search into existing search systems. We mainly encounter these challenges in enterprise search environments which have not kept pace with Web search engines and mostly rely on full-text search systems. Full-text search systems are keyword-based and suffer from the well-known vocabulary mismatch problems. Ontologies model domain knowledge and have the potential for use in understanding the unstructured content of documents.

In this thesis, we investigate the challenges of using domain ontologies for enhancing search in enterprise systems. Firstly, we investigate methods for annotating documents by identifying the best concepts that represent their contents. We explore ways to overcome the challenges of insufficient textual features in lightweight ontologies and introduce an unsupervised method for annotating documents based on generating concept descriptors from external resources. Specifically, we augment concepts with descriptive textual content by exploiting the taxonomic structure of an ontology to ensure that we generate useful descriptors. Secondly, the need often arises for cross-ontology reasoning when using multiple ontologies in ontology-driven search. Once again, we attempt to overcome the absence of rich features in lightweight ontologies by exploring the use of background knowledge for the alignment process. We propose novel ontology alignment techniques which integrate string metrics, semantic features, and term weights for discovering diverse correspondence types in supervised and unsupervised ontology alignment. Thirdly, we investigate different representational schemes for queries and documents and explore semantic ranking models using conceptual representations. Accordingly, we propose a semantic ranking model that incorporates the knowledge of concept relatedness and a predictive model to apply semantic ranking only when it is deemed beneficial for retrieval. Finally, we conduct comprehensive evaluations of the proposed methods and discuss our findings.

Keywords: Ontologies, Document Retrieval, Ontology Alignment, Document Annotation, Semantic Relatedness, Semantic Document Ranking, Search Performance Prediction.

Declaration of Authorship

I declare that I am the sole author of this thesis and that all verbatim extracts contained in the thesis have been identified as such and all sources of information have been specifically acknowledged in the bibliography. Parts of the work presented in this thesis have appeared in the following publications:

- Nkisi-Orji I., 2016, Semantic information retrieval for geoscience resources: results and analysis of an online questionnaire of current web search experiences. Nottingham, UK, British Geological Survey, 15pp. (OR/16/047)
  (**Chapter 3**)

- Nkisi-Orji I., Wiratunga N., Hui K.Y., Heaven R. and Massie S., 2017, September. Taxonomic corpus-based concept summary generation for document annotation. In International Conference on Theory and Practice of Digital Libraries (pp. 49-60). Springer, Cham
  (**Chapter 5**)

- Nkisi-Orji I., Wiratunga N., Massie S., Hui K.Y., and Heaven R., 2018, September. Ontology alignment based on word embedding and random forest classification. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer, Cham
  (**Chapter 4**)

- Nkisi-Orji I., Wiratunga N., Hui K.Y., Heaven R., and Massie S. Generating taxonomic node descriptors for improved semantic document annotation. Communicated to: International Journal on Digital Libraries (2018)
  (**Chapter 5**)

Acknowledgements

I would like to express my most profound appreciation to my supervisory team, Prof. Nirmalie Wiratunga, Dr Stewart Massie and Dr Kit-ying Hui from Robert Gordon University (RGU) and Rachel Heaven from British Geological Survey (BGS). You always offered me the best support and advice whenever I needed them. I am also grateful for the critical and robust supervision you have provided for my research - thank you very much.

This research benefited from generous funding from RGU and BGS through the IDEAS research institute and the BGS University Funding Initiative (BUFI) respectively for which I am very grateful. In the same light, I wish to further thank RGU for offering the training and infrastructure to conduct research, and travel opportunities to participate at conferences and other related events. And to BGS who helped immensely through relevant activities and resources, access to computing resources, and an internship. My sincere gratitude goes to the staff at BGS who have provided expertise, support and participated in surveys while completing this research. Your contributions have helped immensely in my development as a researcher.

I appreciate the insightful feedback received from fellow research students and staff members from the AI research group and SICSA who provided forums for personal development. I also appreciate the help and support from the administrative staff at the School of Computing Science and Digital Media.

Finally, I would like to express my gratitude to my family for their constant love, support and encouragement.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

The Semantic Web has accelerated the development of standards and technologies to enable digital content managers to add meaning to Web documents. The Semantic Web is an extension of the world wide web (WWW) and was born out of the need for a universal framework for data sharing and reuse. Coined in 2001 by Tim Berners-Lee, the Semantic Web has the vision of unambiguously describing Web content thus facilitating machine access to support "intelligent agents". Ontologies are used to capture and organise knowledge, and this forms a vital component of the Semantic Web.

In addition to ontologies, most of the underpinning ideas of the Semantic Web have earlier origins. Enterprise search systems, which are search applications of organisations, have long relied on ontological resources (e.g. classification schemes, subject headings, controlled vocabularies, thesauri) for indexing and knowledge organisation. Medical Subject Headings (MeSH) and Library of Congress Subject Headings (LCSH) are popular ontological resources that have been long used in biomedical and library systems respectively to organise information and to facilitate search and browse.

This work focuses on the role of ontological resources for information retrieval in enterprise search systems. Enterprise search systems had mostly existed on intranet systems but are now increasingly becoming part of the WWW. As a result, the line between the Web and enterprise search systems have become increasingly blurred. While the Web was initially conceived to comprise of documents interlinked by hypertext links, there now exist document collections in multiple formats that are without hyperlinks. Importantly, a significant proportion of documents on enterprise systems have no hypertext

links on which several search algorithms rely on for implementing retrieval strategies. As a result, search in enterprise systems is considered to be more complex than on the Web and several advances in Web search engines are not directly applicable (Hawking, 2004; Li et al., 2014). Ontologies have the potential to bridge the semantic gap between user information needs and the target resource being searched. The importance of the semantic knowledge of ontologies is especially useful for the specialised domains where some terms are not in everyday language use. The ontology forms a semantic layer between the user and target resources to understand domain-specific terms better.

Despite the increase in the availability of domain ontologies and their use to index resources in enterprise systems, effectively utilising the semantic knowledge in ontologies for search remains an open challenge. Most of the work done on the use of ontologies to improve the semantics of search focused on query reformation only (Dalton et al., 2014; Xiong and Callan, 2015).

1.1 Related Research Areas

1.1.1 Ontology

An ontology is "an explicit specification of a conceptualisation" (Gruber, 1993). Conceptualisation refers to an abstract model of some real-world domain. It is an explicit specification because it uses unambiguous language such that it is universally understood. The main components of ontologies are:

- **classes**: concepts or kinds of things in the domain,
- **relations**: specify how classes relate to each other,
- **properties**: features or attributes of classes, and
- **instances**: individuals or objects of classes.

We adopt a general notion of ontology which consists of a broad range of knowledge resources. Accordingly, we refer to knowledge organisation systems such as thesauri and controlled vocabularies as ontologies. The presence of classes (concepts or semantic entities) is a unifier of different types of ontological resources. At times, ontologies do not specify class properties, instances or even relations between entities.

The Ontology Spectrum

As pointed out in the preceding, what constitutes an ontology covers a broad spectrum with varying degrees of specification or formalisation. The ontology spectrum by Lassila and McGuinness (2001) lays out a wide range of what can be considered an ontology as shown in Figure 1.1.

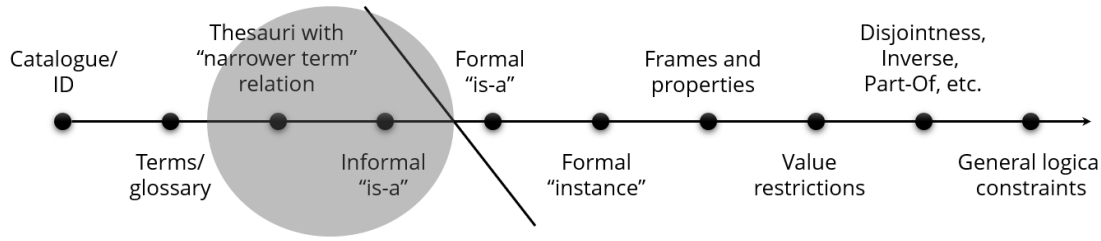


Figure 1.1: Ontology spectrum from lightweight to heavyweight ontologies.

The ontology spectrum in Figure 1.1 range from the simplest notion of an ontology, a catalogue, consisting of a systematic list that unambiguously represents items to very formal constructs that impose logical constraints on the entities they specify. The left side of the spectrum represents knowledge-light (lightweight) ontologies while the right side represents knowledge-rich (heavyweight) ontologies. Knowledge organisation or information retrieval systems commonly use ontologies in the mid-lower range of the spectrum (that is, Thesauri and Informal "is-a"). These ontologies use the specify taxonomic structures using broader/narrower relationships or informal "is-a" relationships. Examples include the MeSH, LCSH, and UNESCO Thesaurus. We use ontologies in the mid-lower range of the ontology spectrum (shaded portion of Figure 1.1) in this work. Specifically, what we refer to ontology consists of knowledge resources that specify semantic entities (or concepts) and subsumption relations between its entities forming a hierarchy of concepts.

1.1.2 Information Retrieval

Information retrieval (IR) deals with finding resources that are relevant to information needs. Information needs are usually expressed as queries using keywords or well-formed questions, and the target resources can be in any media (e.g. text, image, sound). We are particularly interested in document retrieval, which is a form of information retrieval that deals with finding information in unstructured texts. Document retrieval can involve

the search for information in documents or search for the information-bearing documents themselves. The essential components of an IR system for document retrieval are the query input, document collection, matching rule, and search output. The matching rules for determining which documents are relevant or not relevant to a query depends on the representation of documents and queries. Classic document retrieval techniques use keyword-based representations which match query terms against an index of terms of a collection to determine the relevant documents. Relevant documents are expected to contain the query terms.

Information retrieval on the World Wide Web (or the Web) deals with a wide variety of information needs on a large scale. Several advances in web search include search result diversification (Santos et al., 2010), personalisation by creating and utilising user profiles (Hawalah and Fasli, 2015), and training models that learn to rank (Liu, 2011). These advances are due to unclear intents of search queries and in recognising the difference between the ways users search. The search intent behind queries is often difficult to determine which makes meeting information needs difficult. Three commonly identified types of information needs behind search queries are navigational, informational, and transactional needs (Broder, 2002). In navigational queries, the search intent is to reach a specific web site/page that is known to the user. For example, the query "facebook" being issued to get to "www.facebook.com". In a transactional query, the intent is to complete a transaction such as to buy a product or download a file. The search intent in informational queries is to find resources that will provide knowledge about the issued queries. The queries that are issued for document retrieval can be classified under informational queries.

Search intent of informational queries can also be specific or exploratory. With specific search intent, an information need can be met by returning a very relevant document or generating answers to queries posed as natural language questions. Question Answering (QA) is a separate sub-area of IR which deals with answering natural language questions. QA techniques range from using question templates and extracting relevant portions of documents (Andrenucci and Sneiders, 2005) to using knowledge bases (Yih and Ma, 2016) to machine learning techniques using neural networks (Xiong et al., 2016). In exploratory search, the search intent is to obtain a collection of relevant resources (Marchionini, 2006). This type of search deals with more uncertainties, such as whether the query adequately represents the information need and the exact end goal for search. When we discuss search in this work, we are particularly interested in exploratory search in enterprise systems. As discussed earlier, advances in search in enterprise systems lag behind search

on the Web.

1.1.3 Semantic Information Retrieval

Semantic IR techniques explore search methods that can retrieve documents based on meaning. Accordingly, either or both queries and documents are abstracted to high-level concepts (or semantic entities) which capture meanings. The matching rule for retrieval uses the conceptual representations to identify relevant documents even when they do not contain query terms. Ontological knowledge resources are used to model the concepts, and in addition to helping in the matching process, the semantic entities can provide useful information to users.

On the Web, interesting developments in the use of semantic resources for search include the use of Knowledge Graph (Singhal, 2012) and the introductions of Bing Satori (Qian, 2013) and Yahoo Knowledge (Blanco et al., 2013). These ontological knowledge resources specify hundreds of millions of entities and relationships between the entities. Google's Knowledge Graph is perhaps, the most popular ontological knowledge resource for semantic web search. At the time of launch, the Knowledge Graph reportedly had over 500 million entities and 3.5 billion properties of the entities. The entities generally relate to familiar real-life entities such as people, places, organisations, events, movies and music, and are used in different ways to help users meet their information needs. For example, the properties of entities are used to display relevant summaries about the entities being searched. It is not clear how the entities are used to influence the matching process when retrieving documents. Freebase (Bollacker et al., 2008), Wikipedia and the CIA World Factbook are major sources of data for the Knowledge Graph, and entities present in search query logs drive its refinement. The entities cover only the common entity types as they are expected to be most helpful to users, and the semantic knowledge is used to support queries with limited complexity and limited set of recognised terms/entities (Uyar and Aliyu, 2015).

More specialised concepts are encountered in the semantic resources of specific domains. Due to their specialised nature, automatically generating domain-specific semantic knowledge resources is challenging. However, the use of human-defined (or explicit) concepts in ontologies is especially useful for retrieval in specific domains as domain restriction minimises the limitations of using ontologies (Chauhan et al., 2013). For example, domain restriction makes it is easier to maintain domain ontologies and minimises ambiguity in

word sense.

1.1.4 Semantic Resources in Geoscience Information Retrieval

Domain ontologies (e.g. MeSH, Gene ontology) are extensively used to mediate users' queries in the biomedical domain with most works focusing on entity linking and the use of biomedical ontologies for query expansion (Rivas et al., 2014). Most other domains also have ontologies for different applications which include ontologies to index resources and support search. The Linked Open Data Cloud[1] project currently holds over 1,230 ontologies (as of September 2018) across multiple domains ranging from media to government and social networks. We focus on the geoscience domain in this research as it is part-funded by the British Geological Survey[2] (BGS) through the BGS University Funding Initiative (BUFI)[3], and this collaboration enables access to domain resources and expertise. However, we adopt approaches that can be generalised for other domains.

Like most specialised domains, the geoscience domain identifies the role ontologies play in enhancing information use and exchange. This includes the need to achieve geosemantic interoperability and to resolve different representations of geoscience concepts (Reitsma et al., 2009). An example is the use of stratigraphic[4] ontologies to determine the geological age covered by documents (Huber and Klump, 2015). In determining stratigraphic coverage, documents are annotated with entities from stratigraphic ontologies, followed by a resolution of geological age using ontological knowledge. Knowing the geological coverage of documents can provide useful summaries to users and mediate search since various terms refer to similar geological times. Also, there has been work on the use of WordNet to expand geographical terms in queries by adding terms from synonymy and meronymy relations in full-text search (Buscaldi et al., 2005), and to overcome heterogeneity in catalogues (and metadata) and queries by entity linking (Bernard et al., 2004).

Related knowledge resources in the geoscience domain include:

- GeoRef geoscience thesaurus which is used to index material and provide faceted

[1] https://lod-cloud.net

[2] The British Geological Survey conducts research, maintains and generates data, and provides expert services in all areas of geoscience https://www.bgs.ac.uk/about

[3] BGS University Funding Initiative: https://www.bgs.ac.uk/research/bufi/home.html

[4] Stratigraphy relates to the study of rock layers.

navigation feature in the GeoRef database of geoscience publications[5].

- GeMPeT (geoscience, minerals and petroleum) thesaurus[6] which provides a standardised terminology for indexing the geoscience-related materials.
- BGS linked data[7] provides a range of knowledge resources including classification of earth materials, a lexicon of rock units, and geochronology and chronostratigraphy divisions as linked data.
- SWEET (Semantic Web for Earth and Environmental Terminology) ontologies[8] which specify upper-level concept in Earth system science (Raskin, 2006).
- LinkedEarth[9] which is used to organise Earth science data as part of the EarthCube projects initiated by the National Science Foundation (NSF)[10].

As part of this research, we surveyed search application users in the geoscience domain on search experiences and use of semantic resources for information retrieval (Nkisi-Orji, 2016). The responses indicated that many users rely on ontological knowledge for search and will welcome search tools that integrate the semantic knowledge of ontologies in the retrieval process. One use case involves performing multiple searches to retrieve documents for equivalent terms (or alternative spellings) or narrower/child terms of the search intent. Such use case requires having a good knowledge of the content and structure of ontologies to choose the right terms. Not much work has been done on using ontologies to support document retrieval beyond faceted search and query reformulation.

1.2 Research Motivation

Although ontologies have been traditionally used to index resources in enterprise systems, most enterprise search systems use keyword-based search for retrieval. Keyword-based search approaches such as the vector space model and Okapi BM25 are robust and relatively easy to implement over large collections. However, the assumption of term independence in keyword-based search techniques are well-discussed limitations in the literature. The variation in natural language word use between writers allows for the expression of

[5]https://pubs.geoscienceworld.org/georef
[6]http://www.dmp.wa.gov.au/Geoscience-Thesaurus-GeMPet-1564.aspx
[7]British Geological Survey (BGS) linked data: http://data.bgs.ac.uk/
[8]NASA's SWEET ontologies: https://sweet.jpl.nasa.gov/
[9]LinkedEarth: http://linked.earth/
[10]EarthCube: https://www.earthcube.org/info/about

similar ideas in a variety of ways using different terms. Concept-based retrieval has the potential to overcome this challenge because variations and relationships of key terms are captured using high-level concepts. Ontologies explicitly specify such concepts and present two main challenges when used for concept-based retrieval. First, either or both document terms and query terms must be linked to concepts in an ontology. Entity linking (or semantic annotation) cannot be done manually in medium to large systems hence, the need for automated or semi-automated entity linking approaches. Second is the design of effective retrieval techniques that can utilise conceptual representations of documents and queries to meet information needs. Enterprise systems can comprise a variety of resources utilising diverse knowledge organisation systems or ontologies. Achieving an integrated concept-based search system requires the ability to reason across the ontologies of overlapping domains. Reasoning across ontologies requires establishing semantic correspondences between the entities of different ontologies through ontology alignment. Predominant ontology alignment techniques rely on the features of knowledge-rich ontologies and are limited in the types of correspondences they can discover. Accordingly, this thesis investigates the following research questions (RQ1-3):

1. Can the limitations of lightweight ontologies be effectively overcome to discover different types of alignment correspondences?

2. How can lack of sufficient descriptive features in lightweight ontologies be addressed when semantically annotating documents?

3. How can the conceptual representation of documents be exploited to enhance search performance?

1.3 Research Objectives

In order to address issues raised in the use of ontologies for retrieval in enterprise systems, this thesis has five main objectives as follows:

1. Develop effective methodologies for the alignment of knowledge-light ontologies with the ability to discover semantic correspondences.

   This objective addresses RQ1. We are keen to reduce the burden on knowledge acquisition by exploring approaches that rely on using minimal information from

ontologies. However, we must on balance also consider the fact that knowledge-light ontologies often lack the rich features of heavyweight ontologies which most alignment systems rely on.

2. Propose a framework for the semantic annotation of documents that can deal with sparse descriptive textual features in lightweight ontologies.

   This objective addresses RQ2. There is a shortage of benchmark datasets (especially in the geoscience domain) for evaluating semantic annotation systems. Accordingly, we will explore ways to deal with the lack of well-annotated evaluation datasets by making use of readily available datasets (from outwith geoscience) to study the generalisability of proposed methods.

3. Develop a novel semantic ranking algorithm that maximises use of domain knowledge in ontologies for document retrieval.

   This objective addresses RQ3 in part. Key to addressing this objective is to propose methods that can utilise semantic relationships captured in ontologies. We will investigate query and document semantic representational structures by considering the constraints of annotating large document collections which are often encountered in retrieval environments.

4. Investigate use of supervised machine learning to predict when semantic ranking will be beneficial for document retrieval.

   This objective addresses RQ3 in part. The idea here is to avoid the use of semantic ranking in situations where conventional ranking is sufficient. For this purpose, we will investigate the use of features that can facilitate a classifier to differentiate between when to use and not to use semantic retrieval.

5. Propose a semantic document retrieval framework which integrates the semantic ranking model in 3 and the predictive model in 4 above.

   This objective also addresses RQ3. Key to addressing this objective is to propose methods to integrate a semantic component or layer to existing search systems. We are also keen to discover suitable methods and establish suitable datasets for evaluating such semantic document retrieval frameworks.

1.4 Contributions

We describe the contributions of the work in this thesis with respect to the ontology-driven information retrieval stack in Figure 1.2. When using multiple ontologies for semantic IR, **ontology alignment** enables cross-ontology reasoning and ontology merging so that we can treat multiple ontologies as a unit. **Semantic annotation** links the unstructured textual content of queries and target resources to the entities specified in the ontologies. This entity linking task achieves a conceptual representation of resources in the retrieval environment. In the **semantically enhanced IR** layer, conceptual representations are used to enhance search.

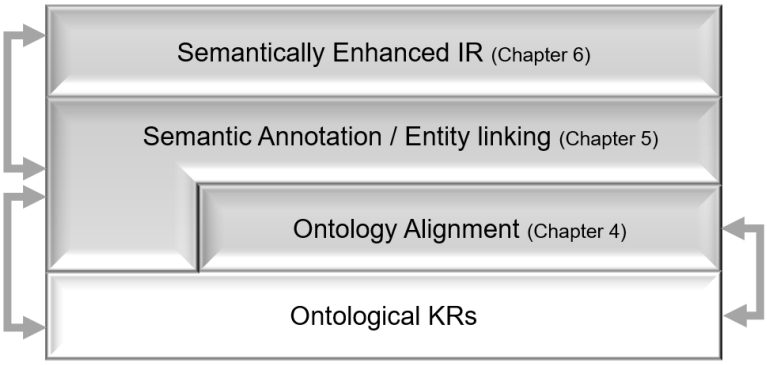


Figure 1.2: Ontology Driven Information Retrieval stack.

We describe our main contributions in the rest of this section. In each contribution, we assume the existence of lightweight ontological knowledge resources with taxonomic structures.

1.4.1 Ontology Alignment

The first contribution is the introduction of novel supervised and unsupervised ontology alignment approaches which integrate string-based and semantic similarity techniques. While string-based and semantic matchers for ontology alignment exist, effectively combining them in an ontology alignment system remains a challenge. Our approaches incorporate word embedding to enhance the discovery of semantic correspondences. A variety of similarity features are used to generate feature vectors for the supervised approach.

A machine classifier uses the feature vectors to align ontologies. The unsupervised approaches use hybrid similarity models integrating string-based similarity, semantic similarity using word embedding, and term weighting components to align the concepts of ontologies.

1.4.2 Semantic Annotation

A second contribution is the development of a framework for the semantic annotation of documents. Linking free text in documents to formal concepts in ontologies is a crucial step in content enrichment, resource linking and information retrieval in general. We link segments of documents to ontology concepts using a corpus-based approach that first enriches concepts using external resources. Being lightweight (or knowledge-light), concepts often lack sufficient textual content for effective use in semantic annotation. It is a two-step process where we utilise knowledge of the semantic neighbourhood of concepts to enrich them with descriptive textual features which we call concept summaries. We then employ a retrieval-based approach and an explicit semantic analysis approach to annotate documents with concepts using the concept summaries. We also introduce a semantic evaluation method for document annotation. Instead of the binary decision on whether the concepts that are returned by an annotation system are correct or incorrect using the standard evaluation method, our semantic evaluation method determines correctness by the degree to which the returned concepts are semantically close to the correct concepts. We expect the semantic approach to give a better indication of the quality of concepts that an annotation system returns.

1.4.3 Ontology-driven Search

The third contribution of this thesis is a SelecTive Ontology-based Retrieval Model (STORM) which selectively use concepts in domain ontologies to enhance search performance in document retrieval in terms of precision, recall and ranking quality. STORM decides whether to use or not use concepts to enhance search based on predicted benefit. A machine classifier uses pre-retrieval query features to predict whether semantic retrieval will improve the retrieval performance for each retrieval task. While concept-based semantic retrieval improves document ranking for some queries, it makes no difference for other queries when compared with retrieval systems with no semantic components. In fact, in some instances, concept-based retrieval worsens the ranking of search results, and

this observation informs the motivation for STORM's design. Considering that concept-based retrieval adds processing and time overheads to the retrieval pipeline, it is desirable to avoid concept-based retrieval when it will not improve the search results. Also, we investigate the use of pre-retrieval query features which ensures that the features for prediction are readily available before any documents are retrieved, and this improves efficiency. We also perform a detailed evaluation comparing alternative concept-based retrieval methods and show how our predictive model can benefit concept-based document retrieval.

1.5 Thesis Outline

An outline of the rest of this thesis is as follows.

Chapter 2 reviews relevant works in literature. We discuss the components of an ontology-based retrieval model and the considerations for different approaches when implementing each component with its limitations. We also discuss relevant works on the annotation of documents using ontological concepts and the alignment of ontologies.

Chapter 3 presents background work which is the motivation for the work in this thesis and approaches adopted. We present the findings from a survey on search techniques and reception of semantic search features. We completed the survey in the initial stages of this research.

Chapter 4 we investigate the problem of ontology alignment, and present novel alignment approaches which aim to enhance the discovery of semantic correspondences during alignment while relying on minimal information in the ontologies. The alignment approaches discussed include supervised and unsupervised alignment which are suitable for different alignment settings. We present an experimental evaluation using the Ontology Alignment Evaluation Initiative (OAEI) benchmark datasets to analyse the ability of the alignment approaches to discover different types of alignment correspondences. We also demonstrate how the alignment approaches introduced compare with state-of-the-art alignment systems which rely on the features of knowledge-rich ontologies.

Chapter 5 explores entity linking for documents and presents a semantic document annotation approach for the conceptual representation of the textual features of documents. We discuss how the lack of descriptive textual features in knowledge-light ontologies affect annotation performance when automating the semantic annotation process. We

demonstrate the augmentation of concepts using externally sourced textual features and explore two approaches for using the generated textual features for semantic annotation. This chapter also introduces an approach for the semantic evaluation of annotation systems and discusses an experimental evaluation which analyses the impact of different considerations in the annotation process.

In chapter 6, we present a semantic document retrieval framework that demonstrates different components of an ontology-based document retrieval model for enhancing search performance. We emphasise the components for semantic document ranking using the conceptual representations of documents and queries. We introduce a predictive model which predicts when semantic ranking will be beneficial for a retrieval task. An evaluation using TREC genomic dataset and the MeSH ontology demonstrates the utility of both the semantic ranking model and the predictive model for document retrieval.

We conclude in chapter 7 with a summary of our contributions and a review of the extent to which we met our research objectives. We also outline the limitations of the work presented in this thesis and considerations for future extensions.

Chapter 2

Literature Review

Ontologies model domain knowledge and enable applications which use ontological knowledge for reasoning. One area of application is semantic information retrieval where ontologies form an explicit semantic layer between users' queries and target resources. The ability to effectively harness the semantic layer for document retrieval remains a challenge. It requires linking unstructured content to ontology entities and using ontological knowledge for retrieval. The needs of specific applications mostly drive the creation of ontologies. Hence, ontologies often specify entities for specific sub-domains only. Achieving broader domain coverage requires merging or integrating multiple ontologies of overlapping domains. The preceding highlights three main areas of interest which we review in this chapter.

1. Ontology alignment which discovers correspondences between the entities of different ontologies.
2. Semantic annotation (or entity linking) which maps unstructured content to ontology concepts.
3. Ontology-driven search which uses conceptual representations to influence search through the semantic ranking of documents.

2.1 Semantic Document Retrieval

The problems associated with treating queries as keywords to be matched in documents for retrieval is well-discussed in the literature (Krovetz, 1997; Krovetz and Croft, 1992). The typical keyword-based approach using the vector space model for full-text search indexes documents by a collection's vocabulary. The index structure is represented as an inverted index which points from each word to the documents that contain the word. Subsequent search and retrieval rely on the lexical matches between the keyword index and the words in a query. Considering that words in a document have different importance, weighting schemes such as the term frequency-inverse document frequency (TF-IDF) are used to reflect the importance of each word in a document. The term weights form vectors in the Vector Space Model (VSM) such that the similarity between the query vectors and document vectors determine the relevance of documents (Manning et al., 2008; Zobel and Moffat, 1998). A major limitation of keyword-based approaches is that they treat words as independent terms and therefore, do not consider the presence of synonyms and polysemous words. Synonyms are similar meaning words which are lexically dissimilar. Keyword search omits relevant documents that contain synonyms of the search terms only. Polysemous words, on the other hand, are words that have multiple senses. In other words, they are different words which are lexically similar. Since keyword-based search considers words based on their lexical forms only, search terms with multiple senses may lead to retrieval of documents containing terms with senses that are different from the search intent.

2.1.1 Retrieval by Query Expansion

In order to alleviate the problems associated with keyword-based search, query expansion approaches attempt to make queries more expressive by adding terms that are deemed relevant to an information need (Carpineto and Romano, 2012). Relevant terms that are used to augment queries are retrieved from a knowledge resource or are discovered by distributional approaches based on the patterns of term use in a document collection. By adding relevant terms to the original query, the expectation is that the retrieval system returns additional relevant documents.

Distributional query expansion approaches use word co-occurrence information to determine expansion terms. Words that often co-occur in documents are assumed to be related. Two main techniques used are global feedback and local feedback (Xu and Croft,

1996). The global technique analyses an entire corpus to discover term relationships. In contrast, the local feedback only analyses the top-ranked documents that the retrieval system returns for the original query. As one will expect, local feedback is sensitive to the number of documents analysed and the proportion of relevant documents in it. However, at its best, local feedback approach performs better than global analysis. In either case, there is also the problem of deciding the number of terms to add to the original query during expansion.

Query expansion using a knowledge resource involves mapping query terms to an ontology or lexical database such as WordNet[1]. Next, the query is reformulated by adding related terms as specified by the knowledge resource. Related terms which are added to a query can be any combination of equivalent terms (synonyms), broader terms, or narrower terms (Bhogal et al., 2007). Broader and narrower terms are identified using taxonomic or subsumption relations. Common challenges in using query expansion include how to determine the extent for expanding terms (e.g. how many ancestors should be added?) and how to determine the importance of added terms (e.g. should added terms have equal weight as the original terms?). Hersh et al. (2000) performed a comparative analysis of different ontology-based query expansion strategies on a biomedical dataset and found that query expansion using synonyms only outperformed expansion approaches which introduced terms from taxonomic relations.

In general, query expansion increases the number of relevant documents retrieved (increase in search recall) by matching alternative query terms and related terms in the target collection. However, query expansion can increase query ambiguity by introducing irrelevant or polysemous terms. Ambiguous queries increase the likelihood of returning more irrelevant documents (decrease in search precision), known as topic or query drift (Bhogal et al., 2007). Therefore, the typical overall impact of query expansion is an increase in recall accompanied by a decrease in precision.

2.1.2 Concept-based Retrieval

While query expansion reformulates a query, other semantic IR approaches discover and incorporate knowledge of term relationships further in the retrieval process such as in Generalised Vector Space Model (GVSM) (Wong et al., 1987) and Latent Semantic Indexing (LSI) (Deerwester et al., 1990). They recognise that even when search terms

[1] http://wordnet.princeton.edu

and target documents use different words, documents that contain words which are sufficiently related to search terms can still fulfil an information need. In GVSM, knowledge of term co-occurrence in a document collection is used to establish correlations between pairs of terms. The co-occurrence knowledge is incorporated into VSM so that even when there are no matches between query and document terms, a document can be retrieved because it contains words that closely correlate with the query. On the other hand, LSI uses singular value decomposition (SVD) to uncover semantic concept structures that are implied in document collections. The LSI technique is based on the principle that words with similar meanings appear in similar contexts and can be discovered through usage patterns. Accordingly, LSI generates a term-document matrix (rows of terms and columns of documents) from a collection using a weighting scheme such as TF-IDF. Using SVD, the matrix rows are reduced to preserve a similarity structure among the columns. The LSI technique requires intensive computation for large-scale implementations and the resulting concepts (groups of related terms) are not often intuitive to humans (Honkela and Hyvarinen, 2004).

Alternative concept-based retrieval approaches use human-defined concepts. Explicit Semantic Analysis (ESA) uses an encyclopedic repository such as Wikipedia[2] to map terms to concepts which they describe (Gabrilovich and Markovitch, 2006). ESA is based on the assumption that the rate of co-occurrence of terms in Wikipedia articles reflects the relatedness of the terms. Egozi et al. (2011) proposed an ESA IR approach which uses Wikipedia articles as concepts and represents terms by concept vectors composing of their TF-IDF weights in different Wikipedia articles. Retrieval is done in the concept space by taking the similarity of the conceptual representations of queries and documents. Although the reported results were impressive, the application of this approach in very specialised domains will be limited by the absence of encyclopedic repositories which describe domain-specific terms like Wikipedia.

2.2 Ontology-based Document Retrieval

Ontologies explicitly specify domain concepts and their relationships making them suitable for use in semantic information retrieval. In document retrieval, both query terms and target documents are mapped to ontology concepts, forming an explicit semantic space for retrieval. Conceptual representations, rather than term representations, are

[2] http://en.wikipedia.org

used for determining the relevance of documents to queries. Ontology-based IR approaches differ in how search queries are input and how they achieve conceptual representations of both queries and documents. Accordingly, we review the primary considerations for an ontology-based IR system in the rest of this section, highlighting differences in relevant works.

2.2.1 Query Entry and Representation

Query entry

One of the primary consideration in an IR system is the representation of queries. Queries represent information needs and are expressed in formats that retrieval systems understand. In traditional IR systems, natural language texts represent queries which the subsequent retrieval process treat as keywords. However, the requirement for conceptual representation of query in most ontology-based IR systems places an added requirement. Fernández et al. (2011) described four query input approaches generally used in increasing order of complexity for users as keyword query, natural language query, controlled natural language vocabulary query, and ontology query language query.

Query input using an ontology query language directly retrieves query concepts from an ontology. SPARQL[3] and RQL[4] are examples of commonly used ontology query languages. This approach removes the extra step required to map query to ontology concepts (Castells et al., 2007). However, it places an additional burden on system users to acquire the skills required to issue ontology query language queries. Also, a good understanding of the content and structure of an ontology is required to input queries correctly. The complexity of these requirements makes this approach impractical in most situations.

In query input using controlled natural language vocabulary, users express queries as natural language texts and also include tags which enable the identification of intended query concepts. This approach also requires expert knowledge of the ontologies and correct tags to use for each retrieval instance. PubMed's[5] Automatic Term Mapping (ATM) uses this approach to discover query concepts. A search log analysis showed that PubMed users did not include any tags in up to 90% of searches highlighting the difficulty in its adoption for practical use (Lu et al., 2009).

[3] http://www.w3.org/TR/rdf-sparql-query
[4] http://doc.apsstandard.org/2.1/spec/rql
[5] http://www.ncbi.nlm.nih.gov/pubmed

A less sophisticated query input method allows users to represent queries as natural language texts which are subsequently processed to identify intended concepts. Here, the entirety of a query conveys additional semantic information instead of a mere collection of key terms. For example, consider the query "What are the locations of igneous rocks in the UK?". The concepts that will likely occur in a relevant ontology are those of "igneous rocks" and "UK". However, other query terms such as "what are the" encode additional information which can help determine relevant documents. Handling such information needs is treated in a separate but related IR field of Question Answering which usually requires performing linguistic analysis in order to extract relevant portions of a document that answers the question (Brill et al., 2002).

The keyword query expresses information needs as a collection of key terms and is the least sophisticated query input method from the perspective of users. Most document retrieval approaches extract key terms in natural language queries even when expressed as well-formed questions. The previous query example can be expressed with keywords as "igneous rocks UK". Keyword query input is used by most search systems which consider commonly occurring terms in natural language texts (e.g. "the") as stopwords. The advantage of using keyword or natural language queries is that users are not required to acquire new query input skills or change how they search in order to use a search application. However, the cost of being user-friendly is the increase in complexity for the retrieval systems as they require additional preprocessing steps to identify the ontology concepts expressed in queries.

Conceptual query representation

Successful mapping of input queries to ontology concepts is a necessary step before document retrieval in several ontology-based IR approaches. In query expansion, this enables search terms to be expanded with related terms as specified by the ontology. Queries are often short collections of keywords which makes it difficult to perform comprehensive linguistic analysis in order to extract underlying concepts. Fernández et al. (2011) automatically discovered ontology concepts expressed in free text queries using a query mapping tool PowerAqua (Lopez et al., 2009). PowerAqua breaks queries up to form triples *subject-predicate-object* with predicates as wildcards. Generated triples' templates form queries to ontologies to discover concepts whose triples match the subjects and objects. As an example, the query "locations of igneous rocks in UK" can form triples *<igneous, ?, rocks>* and *<rocks, ?, UK>*. The latter triple will match *<rocks, locatedIn, UK>*

or <*rocks, foundIn, UK*> if present in searched ontologies. It also searches for triples generated for semantic matches to search terms using synonyms obtained from WordNet. After extracting all the concepts that correspond to generated triples, the next step ranks concepts in decreasing order of relevance based on ontology popularity. Concepts from the ontologies that are well-represented among the candidate concepts are ranked higher than concepts from under-represented ontologies. Using the example, both candidate concepts for "rocks" and "igneous" will likely occur in the same geologic ontology making the musical concept "rocks" from a music ontology to be ranked lower. A drawback for this approach is that it may not discover concepts with multi-word labels such as "igneous rock" when considering the words in the phrase separately. Also, the selection of query concepts from the ranked candidate concepts for use in document retrieval is unclear. The ranking process of PowerAqua's is expected to be less effective when all the ontologies are of similar domain and unnecessary when using a single ontology.

Another approach for representing free text queries as ontology concepts generates all possible n-grams from a query and attempts to match them against textual features of concepts (Meij et al., 2011). Starting from longest n-grams to uni-grams, whenever a query n-gram matches an ontology concept label, the n-gram is removed from the original query and n-grams regenerated from the remaining query. Matching begins from the longest n-grams using the heuristic for choosing between mapping to a concept with a longer textual label or mapping to a concept with the sub-string of the longer label. Mapping the piece of text to the concept with a longer textual label is generally considered to be more specific and a better mapping (Castells et al., 2007). For example, mapping to the concept with label "igneous rocks" is specific and more appropriate than mapping to the more general concept with label "rocks". Semantic matching is achieved using the synonyms specified for concepts. Approximate matching by stemming words can be used to avoid mismatches due to word inflexions (e.g. "rock" versus "rocks") (Shamsfard et al., 2006). Stemming maximises concept discovery in free text queries but also increases the likelihood of selecting unintended concepts.

2.2.2 Semantic Document Indexing

Keyword-based retrieval systems represent documents as bags-of-words in by creating a keyword index for search. In contrast, concept-based retrieval approaches index documents in the search space according to the concepts they contain (Castells et al., 2007; Fernández et al., 2011; Shamsfard et al., 2006). An index of concepts enables semantic

document retrieval that uses the conceptual representations of both query and target documents. A bags-of-concepts representation for documents allows for a semantic index in which a concept points to the documents that contain the concept. Concepts provide a high-level representation of information which, when used for retrieval, can overcome the synonymy and polysemy problems associated with keyword search. A semantic index handles the synonymy problem by mapping similar meaning terms (words or phrases) to the same concept. Accordingly, a retrieval system can return documents that use different terminology for the same topic during search even when there are no lexical matches between a query and relevant documents. Also, the ontology specifies the desired senses for polysemous words which minimises ambiguity when matching relevant documents.

Automating the identification of ontology concepts in a document requires information extraction techniques which often involves some form of string matching between textual content of documents and literals associated with ontology concepts. This process is made difficult due to reasons such as polysemy, word inflexion, use of coreference, abbreviations and symbols in natural language texts (Hazman et al., 2012). The challenges associated with correctly identifying the concepts expressed in text documents are well-discussed in the area of semantic annotation (Reeve and Han, 2005; Uren et al., 2006). Considering the challenges of with achieving a conceptual representation of documents, some ontology-based IR approaches rely on the manual assignment of concepts to document which cannot be scaled to large document collections (Paralic and Kostial, 2003). Others ontology-based IR approaches assume the existence of conceptual document representations in proposed methods (Knappe et al., 2007). In Fernández et al. (2011), conceptual representation is achieved by looking up textual labels of ontology concepts in documents to identify matches. In order to improve accuracy when matching concepts, a concept is considered to be present in a document if the document contains both a label of the concept and the concept's *context*. A concept's context is the set of all entities directly link to the concept on the ontology.

2.3 Semantic Annotation

Documents can be annotated at a higher level instead of identifying individual mentions of concepts. Most of the work in semantic annotation focuses on annotating entire documents or sections within them by the concepts which they discuss. This form on annotation requires an overall understanding of documents' contents and the concepts

which effectively represent them. We categorise the popular approaches for annotating documents as either supervised or unsupervised methods as shown in Figure 2.1. The supervised methods reuse the concepts of previously annotated documents that have features which are similar to a document that is being annotated. On the other hand, unsupervised methods do not rely on a pre-annotated corpus when annotating a new document.

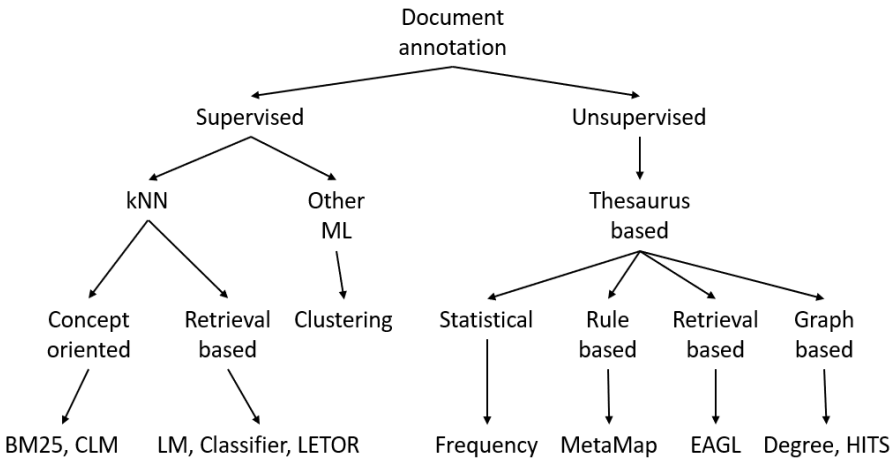


Figure 2.1: Document annotation approaches.

2.3.1 Supervised document annotation

The motivation for supervised approaches is that a target document can inherit some or all the concepts that were assigned to similar previously annotated documents. Concept-oriented approaches generate textual features (or pseudo-documents) for concepts by merging all the documents that have been annotated by each concept. The target document forms a query to indexed pseudo-documents during the annotation process. The concepts whose pseudo-documents are most similar to the target document become the document's annotation. Some approaches for identifying the pseudo-documents which are most similar to target documents are CLM and BM25 (Trieschnigg et al., 2009). CLM uses a language model (LM) for retrieving the relevant pseudo-documents while BM25 uses the Okapi BM25 ranking function (Robertson et al., 1996).

Rather than merge contents to form pseudo-documents, the kNN (k-nearest neighbour) approach indexes each annotated document separately. In order to annotate a new document, a document ranking function retrieves k most similar (or nearest) documents from

the index. The union of all concepts that were used to annotate the nearest documents in the index form candidate concepts for annotating the target document. A variant of kNN ranks candidate concepts by cumulating the relevance scores of all documents in which they form annotations (Giannopoulos et al., 2010; Huang et al., 2011). Other variants of kNN generate and pass features of candidate concepts to a machine classifier which determines the concepts to select (Dramé et al., 2016) or to a Learning to Rank (LETOR) model to rank and select the best concepts (Huang et al., 2011). Some features that are used by a classifier or LETOR include the number of nearest documents that were annotated with a concept and if a concept appears in the title or content of a document. kNN is the state-of-the-art supervised approach and is used in systems such as the Medical Text Indexer (MTI) (Aronson et al., 2004; Große-Bölting et al., 2015). Experimental results show that kNN or hybrids of it are most effective in discovering annotations (Dramé et al., 2016; Trieschnigg et al., 2009). However, we cannot use supervised approaches when a corpus of annotated documents does not exist. It is also difficult to effectively recommend concepts that rarely appear or are absent from the annotated corpus when using the supervised approaches.

2.3.2 Unsupervised document annotation

Unsupervised approaches for annotation rely on the features of concepts in a KR, features of the target document, and external resources. An early annotation system, MetaMap parses a document to be annotated to identify exact and partial mentions of concept terms and these form candidate concepts for annotation (Aronson, 2001). MetaMap ranks candidate concepts using several linguistic principles and selects the best-ranked concepts to form the target document's annotation. Considerations for ranking concepts include the number of times they appeared in a document and whether they were partial or complete matches. The EAGL's approach generates pseudo-documents by merging the textual features of concepts (e.g. concept labels, alternative terms and descriptions). The pseudo-documents form an annotated corpus and the concepts whose pseudo-documents are most similar to a document's content become the document's annotation (Ruch, 2006). EAGL is fast and efficient, but controlled vocabularies often lack sufficient textual content to generate useful textual features. Experiments show that although EAGL outperforms MetaMap, the supervised methods outperform both approaches (Trieschnigg et al., 2009). In MetaMap, the inability to disambiguate terms in documents is one reason for its weak performance (Aronson, 2001).

Statistical approaches rely on direct mentions of concepts in a target document. The likelihood of selecting a concept to annotate a document is proportional to its frequency in the document (Hazman et al., 2012; Medelyan, 2009). Graph-based approaches also rely on direct mentions of concepts which are used to generate co-occurrence graphs (Ohsawa et al., 1998). A co-occurrence graph is constructed for each document by adding directed edges between pairs of concepts that co-occur in the document according to their order of occurrence (Zouaq et al., 2012). For example, given the pair of co-occurring concepts (a,b), each concept forms a node with a directed edge from a to b. Assuming concept c comes after b, the graph is extended by an edge from b to c. We continue the process until we form the entire co-occurrence graph. In DEG, the nodes with the highest degrees (number of edges) are used to annotate a document (Große-Bölting et al., 2015). HITS applies the Hyperlink-Induced Topic Search algorithm to the co-occurrence graphs (Kleinberg, 1999). The HITS algorithm completes an iterative traversal of the graph, computing two scores (hub and authority) for each node. Outgoing edges determine the hub of a node while incoming edges determine its authority. Both scores are summed for each node after the algorithm converges, and the concepts with the highest scores become the document's annotation. A drawback for the statistical and graph-based approaches is that they cannot select a concept that is not explicitly mentioned in a document even when the document has sufficiently discussed the concept.

Also worth mentioning is the annotation of documents using subgraphs of a KR such as DBpedia (Hulpus et al., 2013). First, it identifies the key terms in a document and links the key terms to their corresponding DBpedia concepts. Entity linking returns a subset of DBpedia concepts for each key term through a SPARQL endpoint. Afterwards, a filtering algorithm evaluates the relevance of all the concepts returned and selected one concept for each key term. Titles of DBpedia entries form concept labels while corresponding textual contents provide descriptive textual content for determining which nodes to link to a document. Each DBpedia concept that links to a term in the document forms the root of a subgraph for annotating the document. Subgraphs are extended by exhaustively including all sub-concepts that are relevant to the document. Although reported results are promising, this approach is suitable if the intent is to annotate with DBpedia or similar KR with rich descriptive textual features. When using a different KR, there is the option for using DBpedia as background knowledge. The nodes (concepts) of the KR are mapped to corresponding DBpedia nodes so that the textual features of DBpedia can augment the nodes of the KR. However, in some KRs such as those of very specialised domains, concepts will not have equivalent entries on DBpedia. Also,

DBpedia often conflates terms (e.g. “Rocks” and “Rock type” point to the same article) which is not desirable for maintaining the subtle differences between the concepts of specialised domains.

2.4 Semantic Document Ranking

The interaction between a user and an IR system usually involves the return of rank-ordered documents. Document ranking is especially useful when the target collection contains many documents. It enables users to start assessing search results from the documents which are deemed to be most relevant. With conceptual representations, the matching of query to documents uses the concept space to achieve a semantic ranking of documents. In this section, we review approaches for ranking documents based on the conceptual representation of both queries and documents.

2.4.1 Distances on Hierarchical Paths

Early works on ranking documents with concepts investigated how the distances between query concepts and document concepts correlate with the relevance of documents (Rada and Bicknell, 1989; Whan Kim and Kim, 1990). In Rada and Bicknell (1989), distances on the shortest paths between query concepts and document concepts on MeSH ontology are used to determine the semantic relevance of documents. Let a query, q be represented by a set of concepts $q = \{t_{q1}, ..., t_{qm}\}$ and a document, d be indexed with concepts $d = \{t_{d1}, ..., t_{dn}\}$. A distance function $dist(t_i, t_j)$ between any two concepts t_i and t_j is the number of edges in the shortest path between them on the ontology. The distance between the query and each document is determined as shown in equation 2.1.

$$Distance(d, q) = \frac{1}{n * m} . \sum_{t_i \in d} \sum_{t_j \in q} dist(t_i, t_j) \tag{2.1}$$

Equation 2.1 is the mean of the shortest distances between all pairs of query concepts and document concepts. This method assumes the existence of a taxonomic structure and spreading activation determines shortest path separation between nodes. The spreading activation algorithm begins with a pair of nodes, say a and b. In the first step, we activate nodes that are one hop away from a and b. In each subsequent step, we activate

neighbours of previously activated nodes in an outward direction from each starting node (Collins and Loftus, 1975). That way, the newly activated nodes are a further hop away from each starting node after each step. The process terminates once a node is activated simultaneously from the spreading of a and b. The path that connects a and b through the joint (simultaneously activated) node forms the shortest path from a to b. Computed relevance scores between query concepts and document concepts were shown to correlate well with human judgments of relevance. However, this involved very few documents and relied on the manual annotation of texts. Usually, authors selected one or more MeSH concepts (index terms) to index documents eliminating the need for the automated discovery of document concepts.

2.4.2 Adaptation of the Vector Space Model

Another ontology-driven approach for determining semantic document relevance uses an adaptation of the vector space model (VSM) (Castells et al., 2007; Fernández et al., 2011). This is a numerical statistical approach that adapts the term-based TF-IDF weighting scheme for the weighting of concepts in documents. In the TF-IDF weighting, the importance of a word is proportional to its frequency in the document and is offset by its frequency across the entire collection as shown in equation 2.2.

$$tfidf(t,d,D) = \frac{freq_{t,d}}{max_{t'} freq_{t',d}}.log\frac{|D|}{n_t} \tag{2.2}$$

$freq_{t,d}$ is the number of times the term t occurs in document d, $max_{t'} freq_{t',d}$ is frequency of the most repeated term $t' \in d$, n_t is the number of documents that contain t in the collection being searched, D. In the adaptation for use in concept space, document terms t and t' in equation 2.2 are replaced by document concepts. Assuming $\vec{d_c}$ is the weight vector for the concepts of document d and $\vec{q_c}$ is the vector for the concepts of query q, the semantic relevance of q to d is determined by the cosine similarity of both concept vectors as shown in equation 2.3.

$$similarity(d,q) = \frac{\vec{d} \cdot \vec{q}}{\|\vec{d}\|\|\vec{q}\|} \tag{2.3}$$

The query vectors are either fixed (e.g. set to 1.0 in Fernández et al. (2011)) or supplied by users to reflect the importance of each query concept. A potential drawback for the

adapted VSM approach is that when query concepts are too few, it becomes difficult to differentiate between relevant documents and irrelevant documents. Also, the retrieval process does not incorporate the knowledge of concept relationships as specified by the ontology. Semantically close concepts are placed close to each other in the taxonomic structure of ontologies, and semantic search systems can leverage this semantic knowledge. Accordingly, a solution for very few concepts in queries is to expand the query concepts by adding semantically related concepts. However, but this approach has to overcome the challenges of query expansion.

2.4.3 Enhancing Full-text Search with Ontology

Several research works have explored ways of using ontological knowledge to enhance search effectiveness in full-text retrieval systems. Such works integrate ontology-based semantic relevance approaches into existing document retrieval systems. Ontologies often lack a complete representation of the real-world knowledge required to deal with most information needs effectively. Accordingly, semantic document relevance forms a component of existing retrieval systems by combining semantic relevance with the relevance of the underlying search system. Another motivation for combining the relevance outputs is the knowledge that aggregating the relevance scores of retrieval systems that use different techniques is often better than individually considering the retrieval techniques (Croft, 2000; Lee, 1997). Paralic and Kostial (2003) described a hybrid retrieval system in a Webocracy project which uses the cardinality of the union of query concepts and document concepts to determine the semantic relevance of documents (Paralic et al., 2002). The overall utility of the semantic approach for document retrieval was improved when augmented with either the VSM or latent semantic indexing (LSI) with the VSM hybrid performing best.

Another hybrid of semantic document retrieval uses an aggregation of the concept-based VSM approach and the classic term-based VSM (Castells et al., 2007; Dragoni et al., 2012; Fernández et al., 2011). Fernández et al. (2011) determines the final ranking of documents using a linear aggregation of semantic document retrieval scores and keyword-based documents scores based on Fox and Shaw (1994) as shown in equation 2.4.

$$d_{rank} = \lambda.sem_{rank}(d) + (1 - \lambda).kw_{rank}(d) \tag{2.4}$$

d_{rank} is overall rank of document d, $sem_{rank}(d)$ and $kw_{rank}(d)$ are semantic and keyword-based ranks of d respectively, while $\lambda \in [0, 1]$ is the aggregation weight which determines the contribution of each component.

Analyses of the hybrid approaches show that although adding the semantic component improves the overall retrieval performance, the performance for some queries either remain the same or become worse. The instances of non-beneficial use of semantic ranking suggest that semantic considerations are unsuitable for some queries.

2.5 Choosing Ranking Method by Predicted Benefit

In the previous section, we highlighted that applying semantic ranking does not enhance the performance of all document retrieval instances. We are unaware of any previous works that have considered how to determine queries that will benefit or not benefit from semantic ranking. However, some literature on predicting the performance of queries with respect to one or more retrieval systems are relevant. In query expansion, comparing the language model of documents that are retrieved for a query and language model of documents that are retrieved for the expanded query is used to predict query drift (Cronen-Townsend et al., 2004; Shtok et al., 2009). Also, there has been work on identifying system configurations that are suitable for retrieving documents for different queries and reusing the best configuration identified for a query whenever it is repeated (Bigot et al., 2015). This approach is suitable for search environments where queries repeat regularly. These performance prediction approaches rely on post-retrieval features such as document retrieval scores and query clarity scores which are generated from feedback documents. Post-retrieval features are only available after a retrieval system returns an initial set of documents for a query. Determining these features can be time-consuming making them unsuitable for use in practice in most cases.

In contrast to post-retrieval features, pre-retrieval features are available before a search is performed. Typical examples of pre-retrieval features include query features such as the length of queries and corpus-dependent features such as the inverse document frequency of query terms. Given that pre-retrieval features are determined relatively quickly, they are most suitable for use in practical search systems (Hauff et al., 2008; He and Ounis, 2004). Even better are query features for performance prediction that are both pre-retrieval and independent of the target collection (i.e. the corpus being searched). Corpus-independent features are suitable when there is limited access to the

target collection and have the advantage of ease of transfer to a different collection. Katz et al. (2014) uses corpus-independent features for performance prediction by relying on the features extracted from queries (e.g. query length) and features from Wikipedia (e.g. the number of article titles that match all or part of query terms). Examples of other related works which generate features for queries include features for mapping queries to the concepts of a knowledge resource (Meij et al., 2011), features for generating a document ranking model in learning to rank (Liu, 2011) and features for classifying queries according to users' intentions (e.g. navigational or informational intent) (Figueroa and Atkinson, 2016).

2.6 Semantic Relatedness

Incorporating the knowledge of concept relatedness in an ontology-driven IR system requires an approach for determining semantic relatedness. Semantic relatedness approaches quantify the degree of relatedness between terms or entities. Any concepts that have some commonality in their meanings are semantically related (Budanitsky and Hirst, 2006). Semantic similarity is a more specific form of semantic relatedness which deals with "is-a" type of relationships. Semantic similarity implies semantic relatedness, but semantic relatedness does not always mean semantic similarity. While a "car" is both related and similar to a "motorbike", it is only related to "driving". Semantic distance is the inverse of semantic similarity. Semantic distance increases with decreasing semantic similarity and decreases with increasing semantic similarity. In the rest of this section, we review several approaches for determining semantic relatedness between entities. We adopt the categorisation of methods for determining semantic relatedness as either based on knowledge resource or based on distributional semantics as summarised by Figure 2.2.

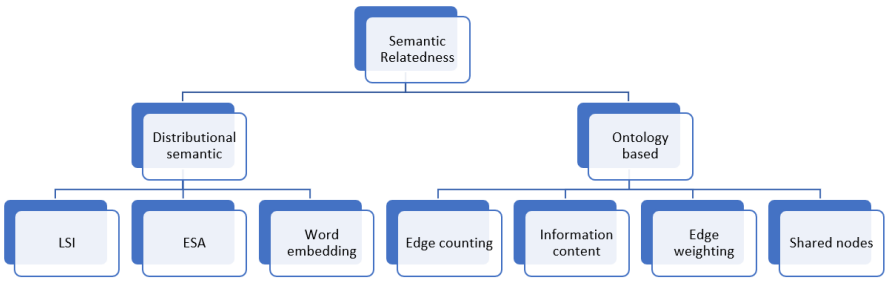


Figure 2.2: Classification of semantic relatedness approaches.

2.6.1 Distributional semantics approaches

Distributional approaches rely on the distributional properties of natural language terms (e.g. words, noun phrases) such as co-occurrence patterns or similarity of contexts in corpora to determine semantic relatedness. These approaches follow the distributional hypothesis which states that terms with similar distributional patterns tend to have the same meaning.

Latent Semantic Analysis

Latent Semantic Analysis (LSA) exploits the co-occurrence patterns of terms in documents to create a semantic concept space (Deerwester et al., 1990). LSA assumes that semantically related terms will often co-occur in similar documents or piece of texts (e.g. paragraphs). It begins by constructing a term-document occurrence matrix with words (or terms) forming rows and documents forming columns. Weighting functions (or matrix elements) for the term-document incidence matrix can vary from binary (1 if the term exists in the document and 0 otherwise) to TF-IDF weights. In a sizeable corpus, this matrix is very sparse with high dimensions considering the vocabulary size of natural language texts. Accordingly, LSA applies singular value decomposition (SVD) to reduce the rows of the term-document matrix while preserving the similarity structure between its columns. SVD decomposes the term-document matrix D into three matrices: a term by dimension matrix U, a document by dimension matrix V, and a diagonal matrix of singular values S as shown in equation 2.5.

$$D = U \times S \times V \tag{2.5}$$

S is rank-truncated to the k most important concepts so that multiplying U by S ($U' = U \times S$) produces a reduced term by document matrix. The cosine similarity between term vectors in U' determines their semantic relatedness.

Explicit Semantic Analysis

Explicit Semantic Analysis (ESA) uses an encyclopedic repository such as Wikipedia to represent terms according to the documents they appear in (Gabrilovich and Markovitch,

2006). Each document represents a concept, and the extent to which terms describe similar concepts determines their semantic relatedness. ESA requires a collection where the documents are dissimilar while maintaining a good cohesion between the terms appearing in each document. An encyclopedic repository is chosen as the corpus for ESA because each document focuses on a topic. As a result, each concept's description is topically orthogonal the description of other concepts.

Given a term t and collection D, the vector representation for t is:

$$\vec{v_t} = \{\text{tfidf}(t, d_1), ..., \text{tfidf}(t, d_N)\} \tag{2.6}$$

$\text{tfidf}(t, d)$ represents the TF-IDF weight of t in each $d \in D$. Equation 2.6 gives a concept vector representation of length N ($N = |D|$) for each term in the vocabulary of D. The cosine similarity between the concept vectors of terms determines their semantic relatedness.

Word embedding

Word embedding approaches map natural language words or phrases from a vocabulary to real-number vectors using language modelling and feature learning techniques. Mikolov et al. (2013b) introduced word2vec, a novel predictive model for learning word embedding from texts and has gained widespread use. Word2vec learns the vector representation of words using a two-layer shallow neural network language model in a computationally efficient manner. The word2vec model has two variants, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model (skip-gram). In generating a word2vec model, the neural network is trained to maximise the conditional log-likelihood for predicting target words from context words for CBOW or for predicting context-words from the target words for skip-gram. The CBOW treats all the context words of a target word as a single observation in the training data while the skip-gram treats each pair of context word and target word as a separate observation. The skip-gram model has proven to be more accurate than the CBOW because it generates more generalisable contexts (Mikolov et al., 2013a). Word2vec models are trained using either or both hierarchical softmax and negative sampling for computational efficiency. The learned representation for words or phrases forms their embedding vectors which are compared (e.g. using cosine similarity) to determine relatedness between terms.

2.6.2 Ontology-based approaches

Several ontologies specify taxonomic (or semantic) relations between concept and other properties. Ontology-based approaches determine semantic relatedness using knowledge of semantic relations between concepts or the similarity of concept properties. In this section, we explore different ontology-based semantic relatedness measures according to the source of evidence used to establish relatedness and other considerations made in each approach. Table 2.1 summarises the features of different ontology-based semantic relatedness approaches. While some approaches such as Rada et al. (1989) and Wu and Palmer (1994) only depend on evidence from an ontology to determine relatedness between its concepts, others like Resnik (1995) require a corpus to measure the information content of concept terms.

Table 2.1: Features of ontology-based semantic relatedness approaches. Corpus indicates that an approach incorporates information from a corpus in determining relatedness.

	Corpus	**Method**				**Symmetry**	**Depth scaling**
		Edge counting	Info content	Edge weighting	Shared nodes		
Rada et al.		✓				✓	
Sussna				✓		*	✓
Wu and Palmer		✓				✓	✓
Resnik	✓		✓			✓	✓
Jiang and Conrath	✓		✓			✓	✓
Leacock and Chodorow		✓				✓	✓
Lin	✓		✓			✓	✓
Hirst and St-Onge				✓		✓	
Knappe et al.					✓	*	✓

Semantic Relatedness Methods

Edge counting Several semantic relatedness approaches establish relatedness measure between concepts using the semantic relations specified by an ontology (Leacock and Chodorow, 1998; Rada and Bicknell, 1989; Wu and Palmer, 1994). Edge counting approaches rely on the number of nodes or edges between concepts. Rada et al. (1989) uses count of edges in the shortest paths between concepts to determine semantic distance. Hence any two concepts with equal distance of separation will have the same semantic distance irrespective of their position on the ontology. Using separation distances alone to determine semantic relatedness does not reflect the nature of most ontologies where, following taxonomic relations, concepts become increasingly specific with increasing depth

(distance from the root node). Accordingly, approaches such as Leacock and Chodorow (1998) and Wu and Palmer (1994) consider the distance of concepts from the root node and they improve on the shortest path separation approach. The relatedness between neighbouring pairs of concepts increases the farther away they are from the root node.

Edge-counting approaches are susceptible to incompleteness or inconsistencies in ontologies. Semantically distant concepts can be placed close to each other because portions of an ontology are incomplete. Also, the decision on which relation types to use in the ontology to determine separation distances and how to assign appropriate weights to edges are mostly subjective. These considerations usually require manual judgements and tuning in order to set appropriate edge weights (Hirst and St-Onge, 1998).

Information content Rather than depend on concept relationships, several semantic relatedness approaches use an external corpus to measure the information content of concepts (Jiang and Conrath, 1997; Lin, 1998; Resnik, 1995). Information content approaches use distributional statistics of terms in a corpus in order to establish relatedness between concepts. The probability of occurrence of a concept and all other concepts it subsumes, in the corpus determines the concept's information content as equation 2.7 shows.

$$P(c) = \frac{freq(c)}{N} \tag{2.7}$$

$freq(c)$ is the combined frequency of concept c and the frequency of the concepts it subsumes in the corpus and N is the total number of concepts in the corpus.

Counting subsumed concepts towards a concept's frequency ensures that concepts which are lower in the hierarchy possess higher information content (Blanchard et al., 2005). By using a corpus, the information content approach is expected to minimise problems associated with inconsistencies in an ontology's design. However, it may not be able to appropriately differentiate between multiple senses of polysemous words in the corpus. For example, it is difficult to differentiate between "rock" – *music genre* or *stone* – in a large document collection without requiring additional natural language processing operations to disambiguate word sense. As a result, the corpus-based can obtain lower-than-expected relatedness values by conflating all senses of polysemous terms in a corpus. Using a domain-specific corpus to determine information content will minimise instances of polysemous terms. Another challenge when using information content approaches is

that they can produce unexpected relatedness measures. As Richardson et al. (1994) demonstrated using the Resnik approach (Resnik, 1995) and WordNet, "Ambulance" compared with "Convertible" was shown to be more related than "Motor Vehicle" compared with "Motor Vehicle". The performance of edge-counting and information content methods are comparable, but on the average, the information content methods perform better (Hliaoutakis et al., 2006).

Edge weighting Edge weighting approaches determine semantic relatedness based on the weights assigned to semantic relations between the concepts of an ontology (Hirst and St-Onge, 1998; Sussna, 1993). In Sussna (1993), the weight of an edge between a pair of nodes (concepts) depends on the uniqueness the edge when compared with outgoing edges from the pair to other nodes. For example, let a relation of type, r form an edge from concept c_1 to c_2. The weight of r for becomes less if there are several other type-r relations from c_1 to other concepts. The assumption is that the strength of r becomes increasingly "diluted" as the number of r emanating from c_1 to concepts that are not c_2 increases. Consider the "has-a" relation used to denote components parts, edge-weighting is appropriate for weighing this edge because the strength of the relationship between an entity and its components is expected to be stronger when it is made up of very few components compared to when there are many components.

Shared nodes Knappe et al. (2007) introduced the shared nodes approach. When comparing two nodes, the shared nodes approach considers the overlap in all nodes that are reachable by the upward spreading of both nodes. A higher overlap in upward reachable nodes implies increased semantic relatedness between nodes. By considering all nodes through all possible paths, it avoids any pitfalls that may arise from considering only one path between nodes. In an ontology of mixed entities (e.g. classes and properties), two classes in different subparts of the ontology can have common (upward reachable) properties which will contribute to a relatedness value that is higher than considering the shortest path between the classes.

Symmetric property

Semantic relatedness approaches are classified as either having the symmetric property or asymmetric property. With the symmetric property, the relatedness between concepts remain the same irrespective of the direction of comparison (i.e. $f(x, y) = f(y, x)$). Most

of the approaches considered in Table 2.1 have the symmetric property. For the edge counting approaches, distances remain the same irrespective of the direction of counting. Knappe et al. (2007) uses two sub-expressions, and its aggregation weight determines its symmetric nature. The overall expression is only symmetric when the sub-expressions are combined equally, but the authors favoured asymmetry for a query expansion use case. Also, Sussna (1993) can be made to be asymmetric if we consider edge weights in one direction for a pair of concepts instead of using an average of weights measured from both directions. The motivation for the asymmetric property is that the perception of relatedness can differ depending on the direction of comparison as pointed out by Tversky (1977). For example, an "ellipse" is more like a "circle" than a "circle" is like an "eclipse". An eclipse may not be one of the first things that come to mind when describing a circle. Therefore, the relatedness of eclipse to circle is higher than the relatedness of circle to eclipse.

Depth-scaling

As earlier mentioned, the relatedness measures of concepts are scaled according to their taxonomic depths on the ontology hierarchy for several semantic relatedness approaches. Typically, concepts become increasingly specific as the taxonomic structure of an ontology is traversed from the root to leaf nodes. Therefore, concepts in close proximity are more closely related when they are nearer the leave nodes than when they are nearer the root. In other words, the cost for traversing nodes is lower for specific concepts than for general concepts. This is known as the specificity cost property (Knappe et al., 2007). Rada et al. (1989) is not depth-scaling approach, and is outperformed by the other approaches in experiments which test for the correlation between semantic relatedness approaches and human judgements of relatedness (Hliaoutakis et al., 2006).

2.7 Ontology Alignment

In systems requiring the use of ontologies, the need often arises to align or merge multiple ontologies of overlapping domains. Ontology alignment, also referred to as ontology mapping or matching, deals with establishing semantic correspondences between concepts of different ontologies (Euzenat et al., 2007). Trends in research output indicate that ontology alignment is an active and growing area of research (Otero-Cerdeira et al., 2015).

The needs of information systems, the Semantic Web and projects such as the Linking Open Data (LOD)[6] are examples of drivers of current interests in ontology alignment. Ontologies differ widely, and as a result, alignment approaches often require bespoke implementations which can perform very well on one alignment task and perform very poorly on another (Jain et al., 2010).

When given a source ontology and a target ontology for alignment, a typical alignment process compares each entity of the source ontology with all entities of the target ontology to determine alignment correspondences. Concept comparison in alignment systems involves the use of one or more matching techniques as Figure 2.3 shows.

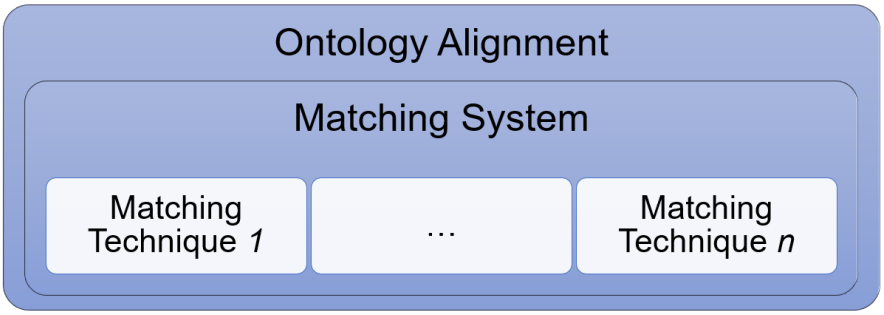


Figure 2.3: Ontology alignment as having a matching system composed of matching techniques.

Accordingly, we categorise research in ontology alignment as:

1. matching techniques which aim to identify strategies and similarity metrics that indicate when entities can align; or
2. matching systems which use one or more matching techniques to align ontologies.

2.7.1 Matching Techniques for Ontology Alignment

Establish an alignment correspondence between a pair of concepts requires comparing the concepts for the relation type of interest. Comparisons for alignment are either element-level or structure-level (Otero-Cerdeira et al., 2015). In element-level comparison, intrinsic features of concepts are used such as string comparison of concept labels. One of the most basic approaches for element-level matching is to compare concept labels for exact string matches. Check for exact matches is unable to deal with spelling

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

differences or word inflexions. Accordingly, most alignment approaches use edit distance metrics such as *Levenshtein* and *Euclidean* to identify close matches in the textual features of concepts (Li et al., 2009). The textual features of concepts include concept labels, property names, comments, notes and definitions. In Li et al. (2009), the textual features of concepts form pseudo-documents for generating TF-IDF vectors. The comparison of the concepts is achieved by the cosine similarity of their vector representations. The textual features of concepts are crucial for successful ontology alignment (Ngo et al., 2011). However, descriptive texts are often insufficient in knowledge-light ontological knowledge resources.

Structure-level similarity considers the neighbourhood of concepts in order to establish a type of alignment correspondence. Even when concepts share little element-level information, alignment can be successfully discovered using structural information. One approach is to compare the ontological neighbourhood (e.g. parent nodes, child nodes, or siblings) of concepts. This approach is based on the expectation that equivalent concepts in different ontologies will have similar neighbours. External resources such as Wikipedia and Google page count are also be used for structure-level matching (Jain et al., 2010; Jiang et al., 2014). Jain et al. (2010) generates a tree of Wikipedia articles for each concept and compares the trees to determine alignment correspondences. In order to generate a concept's Wikipedia tree, the concept forms the root of the tree, and its label forms a query to the Wikipedia Web Service. Titles of returned Wikipedia articles become the first-level nodes of the concept's tree. In the next step, each first-level node becomes the query to Wikipedia, and the titles of returned articles become second-level nodes. The process of using previous titles to retrieve new articles is repeated until the tree grows to a specified level. The degree of overlap of the nodes of concepts' trees determines the similarity between the concepts. The expectation is that similar trees will be generated for similar concepts since article titles are unique on Wikipedia.

2.7.2 Matching Systems for Ontology Alignment

The use of a single matching technique can be unreliable in determining when concepts match (Otero-Cerdeira et al., 2015). For example, a pair of concepts with different textual labels is not conclusive evidence that they cannot form an alignment. As a result, typical matching systems use various strategies to combine multiple matching techniques when aligning ontologies. CroMatcher (Gulić et al., 2016), AgreementMaker (Cruz et al., 2009) and YAM++ (Ngo and Bellahsene, 2012) are examples of state-of-the-art ontology

alignment systems based on their performances in recent Ontology Alignment Evaluation Initiative (OAEI)[7] competitions, and they all use multiple matching techniques for alignment. CroMatcher uses nine matching techniques to compare concepts and has outperformed other alignment systems at several recent OAEI challenges. YAM++ uses multiple similarity metrics in a supervised machine learning algorithm to determine when concepts align. When trained on a dataset, the algorithm selects best features (similarity metrics) to use for alignment. A similarity metric such as string matching between concept labels is not useful for alignment if symbols represent the label of concepts. In such cases when direct comparisons are not informative, structure-level matching techniques are better for comparing concepts. The choice of matching techniques and determining composition weights for multiple similarity metrics have been the subject of multiple research works (Gulić et al., 2016; Martínez-Romero et al., 2013).

2.7.3 Semantic Similarity for Ontology Alignment

String comparisons become less useful for alignment when the vocabulary of ontologies differ. As a result, semantic matching techniques attempt to match concepts by meaning to discover alignments which string-based similarity techniques omit. Some ontology alignment systems use external knowledge resources such as WordNet and Wikipedia to estimate semantic similarities (Husein et al., 2016; Jain et al., 2010; Lin and Sandkuhl, 2008). Using an external resource requires anchoring concepts to the external resource and using the external resource for inferencing. Semantic relatedness approaches are used for measuring similarities on WordNet or similar hierarchical structures. In general, the relative positions of the nodes to which concepts are anchored on WordNet determine their semantic similarity. Approaches for obtaining similarity include edge counting, shared nodes, information content and hybrid methods (Blanchard et al., 2005; Knappe et al., 2007).

Recent experiments show that the use of word embedding vectors (e.g. word2vec) outperforms the use of lexical databases for semantic matching (Zhang et al., 2014). Word embedding preserves several linguistic regularities and similarity between word vectors have been shown to correlate well with human judgements. When a sizeable corpus is used to generate word embedding vectors, its vocabulary coverage is higher than the coverage of current lexical databases. The use of word embedding is also promising for

[7] http://oaei.ontologymatching.org/

cross-lingual alignment by jointly embedding ontologies in a vector space (Sun et al., 2017).

Despite its usefulness, many alignment systems do not use semantic matching techniques because the effective integration of string-based similarity and semantic similarity remains a challenge (Otero-Cerdeira et al., 2015; Shvaiko and Euzenat, 2013).

2.7.4 Alignment Relation Types

While most ontology alignment systems aim to discover equivalent relations between concepts of different ontologies (e.g. "same-as", exact-match, or close-match), the need to discover other relation types such as subsumption relations (e.g. is-a, broader-than) can arise. After establishing equivalent relations, a reasoner can be applied to discover subsumption relations. However, there are situations when the ontologies to be aligned have no equivalent concepts. Such situations require alignment techniques that discover subsumption relations. Also, the discovery of subsumption relations is required when aligning a lower-level ontology (or domain ontology) is to an upper-level ontology. This form of alignment is, at times, referred to as ontology articulation (Tatsiopoulos and Boutsinas, 2009). Spiliopoulos et al. (2010) achieved subsumption alignment using a composition similarity metrics in a supervised machine learning approach. A machine learning algorithm uses the features of known subsumption relations for identifying new subsumption relations. A challenge one may encounter when using this approach is in the generation of a training dataset since most of the existing alignment datasets are on equivalent relations.

2.8 Chapter Summary

In this chapter, we reviewed the literature on the use of domain ontologies for document retrieval. We discussed various components of retrieval systems that use ontologies and highlighted key differences in considerations for each component. Our review of ontology-driven document retrieval includes exploring schemes for the conceptual representation of both queries and documents. The discussion highlighted that making the input of queries easier for users results in increased complexity for mapping queries to the concepts intended by users. On the other hand, placing the burden of identifying query concepts on users makes it difficult for users to construct queries. In the conceptual representation

of documents, we reviewed different annotation approaches and discussed the practicality of each approach.

We also discussed a variety of approaches for the use of concepts for ranking documents after representing both queries and documents as ontological concepts. We showed that the semantic relations between concepts do not influence the ranking process of current ontology-driven document retrieval approaches. Ontologies that have taxonomic structures specify the semantic relatedness of concepts and using the knowledge of concept relatedness is promising for exploratory search. Accordingly, we reviewed ontology-based semantic relatedness approaches according to the properties of their algorithms and highlighted their strengths and weaknesses.

After that, we reviewed matching techniques for ontology alignment and how matching techniques form building blocks for alignment systems. Ontologies differ and are often fragmented, thereby covering overlapping domains. Accordingly, the need often arises to align multiple ontologies for broader domain coverage. We highlighted the need for better integration of matching techniques based on string-based similarity and semantic similarity.

Chapter 3

User Survey and Evaluation Datasets

Ontologies form a crucial component for achieving the Semantic Web vision and play an essential role in information systems for search and information sharing among other uses. Knowledge-light ontological resources in the form of thesauri, knowledge graphs or controlled vocabularies are often used to index resources to enhance organisation and retrievability. Over the years some research works have explored how to leverage ontological knowledge resources to help users to discover the information they seek. In recent times, however, widespread use of full-text search systems for resource discovery makes the integration of ontologies in the retrieval process challenging. Ontologies form an explicit semantic space between an information need and the target resource when used to for search. Both documents and information needs have to be linked to entities in the ontology in order to harness ontological knowledge. Also, the ability to reason across ontologies is expected to be advantageous when multiple ontologies are used in an information system. Accordingly, the following are considered for ontology-driven information retrieval.

1. How to map search queries and documents to ontology concepts.
2. How to exploit semantic knowledge in the ontologies to enhance retrieval performance.
3. How to align multiple ontologies to support reasoning across ontologies and for ontology merging.

3.1 User study on semantic search

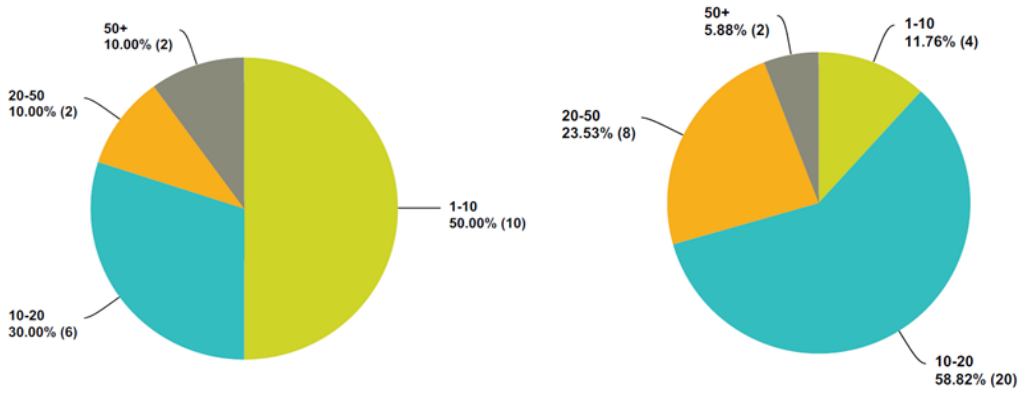
In order to gain better insight into the role of ontologies in document retrieval, we undertook a user study to investigate how ontologies influence search even when they are separate from the search application. The user study was by questionnaire titled "Semantic web searches for geoscience resources" designed to better understand current search habits and preferences, and semantic search considerations in document retrieval (see Appendix B). With this research being a collaboration with the British Geological Survey (BGS), it was relatively easier to find domain experts in the geoscience domain who volunteered to respond to the study. Over a month from July to August 2015, the questionnaire was completed by 35 staff members of BGS over the Internet. The rest of this section presents relevant findings from analysing the survey's responses.

3.1.1 Search results and relevance

Questions were separated according to the broad categorisations of search intent as either lookup search or exploratory search (Athukorala et al., 2016). Lookup search has clear search goals, and the intent is to find a relevant document that meets the information need quickly. On the other hand, the search goals are less clear in exploratory search, and the intent is to find as many relevant documents as possible. As expected, there was a higher tendency for respondents to assess more search results in exploratory search than in lookup search as shown in Figure 3.1.

While about 88% will assess more than 10 results in exploratory search, only 50% will do the same in lookup search. However, in both search categories, most respondents will not assess more than 20 search results. Twenty search results correspond to about the first two search results pages in popular search engines such as Google and Bing search[1]. In lookup search, the respondents are either unwilling to assess many search results or do not need to assess many search results. The latter scenario is a possible explanation since a relevant entry among the first few search results will make assessing additional results unnecessary. Ranking the most relevant document as high as possible is vital in lookup search. In exploratory search, having many relevant documents near the top rankings are essential. Information retrieval techniques such as query expansion are more suited for exploratory search.

[1]There are 10 entries per page using default setting as of 2018



(a) Lookup search (b) Exploratory search

Figure 3.1: Number of results that are assessed for relevance according to category of search.

Furthermore, all respondents reported that their search results were at some point, dominated by irrelevant result entries. Often, an irrelevant document contains the search terms but not used in the intended sense. As seen in Figure 3.2, the domination of search results by irrelevant entries was more than a rare occurrence for 77% of respondents. Only 23% said this seldom happened hence, improving search to avoid irrelevant result entries will be beneficial.

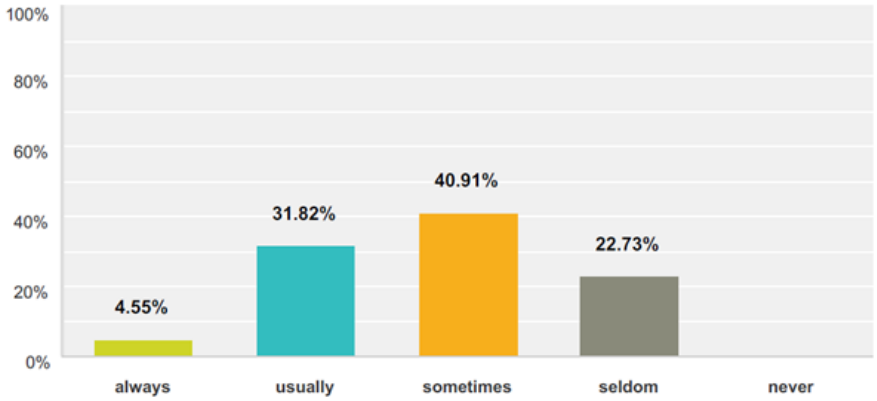
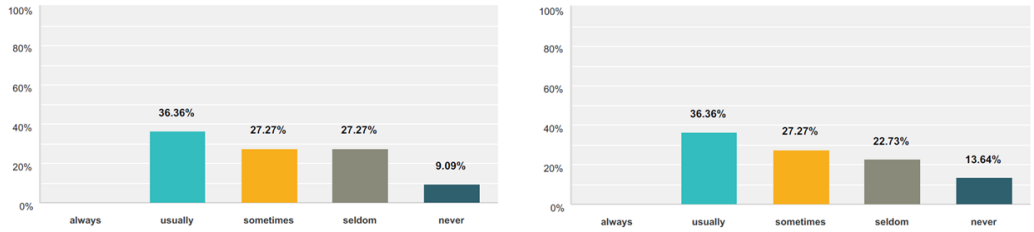


Figure 3.2: Response to how often search results are dominated by entries that are not relevant.

3.1.2 Semantic considerations during search

We asked respondents how often they selected terms from ontological knowledge resources during search and the types of terms selected. The results are summarised in Figure 3.3.



(a) Use of equivalent terms

(b) Use of narrower terms

Figure 3.3: Tendency to perform multiple searches or use advanced search features to include terms from controlled vocabularies.

Eighty-two per cent (82%) usually or sometimes performed multiple searches or constructed advanced search queries in an attempt to include narrower or equivalent terms (or alternative spellings) to original search intent. The respondents provided identical responses to both questions as captured in Figures 3.3(a) and 3.3(b).

3.1.3 Importance of semantic search applications

Ninety-five per cent (95%) of respondents think that a search feature to include narrower or equivalent terms from controlled vocabularies to original search intent is beneficial. Ninety per cent (90%) of those who want narrower or equivalent terms included from controlled vocabularies prefer to have control over its use. In other words, they would like the ability to turn the feature on or off (see Figure 3.4). The other 10% want such feature included implicitly. Forty-eight per cent (48%) of respondents prefer to use such feature by default with the ability to turn it off while 38% do not want it turned on by default. Only 5% of respondents think that such feature is not of benefit to them. Considering that a significant 43% do not want this feature as default search option or do not deem it beneficial, it may be most appropriate to include it as an optional search feature which a user can turn on.

Eighty-one per cent (81%) prefer having the ability to specify the intended context/meaning of search terms but to do so only when such terms are ambiguous. There was strong

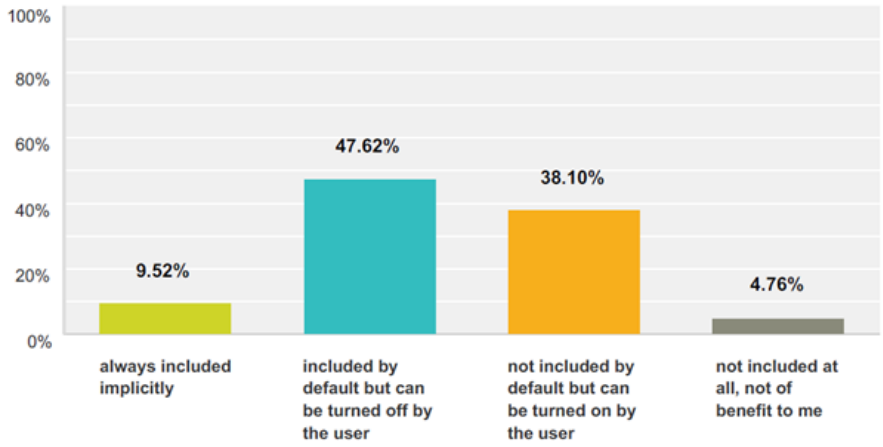


Figure 3.4: Response to how often search results are dominated by entries that are not relevant.

support for a feature that allows users to select from a list of alternative definitions whenever search terms were ambiguous. As shown in Figure 3.5, about 10% want intended context/meaning of search terms to be decided by the search engine. Only 5% thought that a search feature to resolve ambiguity in search terms did not benefit them.

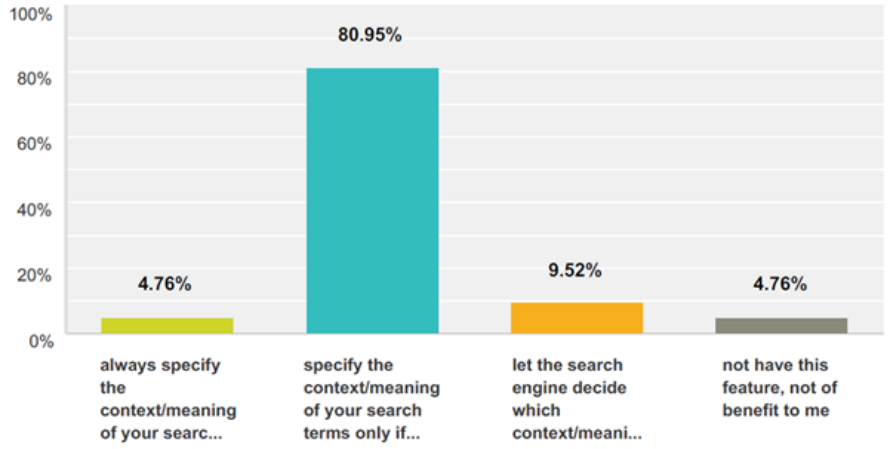


Figure 3.5: Response to how often search results are dominated by entries that are not relevant.

3.1.4 Useful Knowledge Resources for semantic search

The participants of the survey were asked to identify the ontological knowledge resources that are useful to them for implementing semantic search functionalities. Figure 3.6 shows the responses with the percentage of users who chose each knowledge resource.

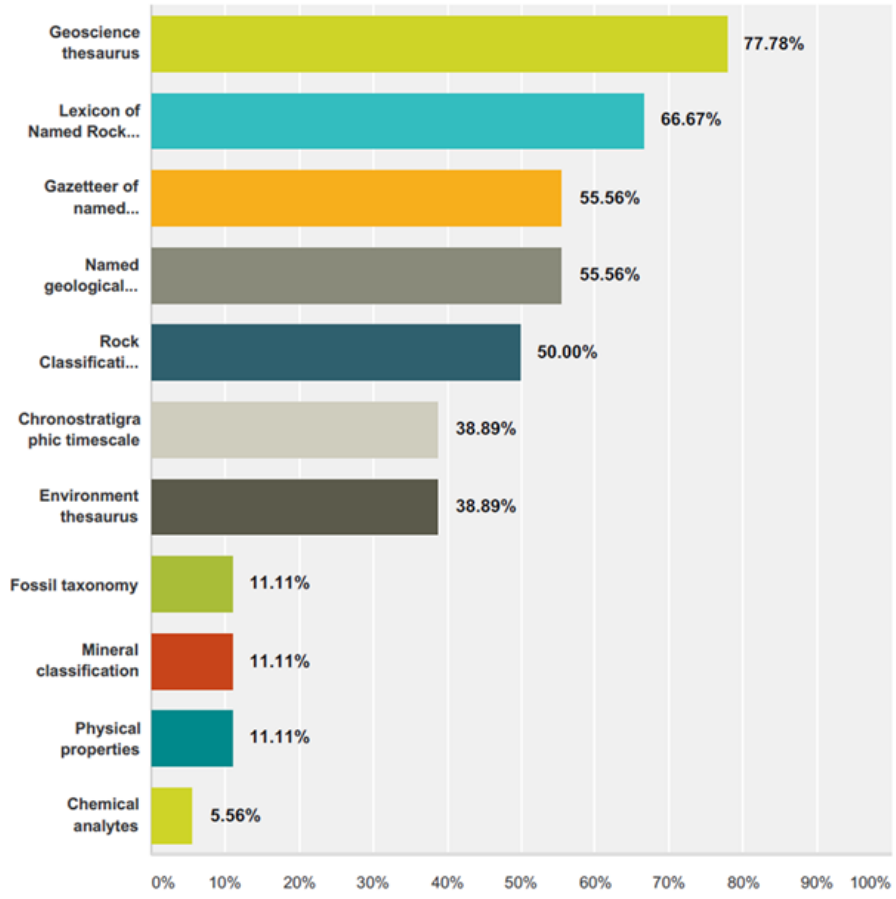


Figure 3.6: Preference of vocabularies to implement semantic search.

Figure 3.6 shows that about 78% of respondents selected the Geoscience thesaurus[2] as a useful resource for implementing semantic search features. The Geoscience thesaurus is a knowledge-light ontological resource which describes general geoscience-related concepts, specifying alternative terms (synonyms and spelling variations) and taxonomic relations. This popularity of this thesaurus was expected as it is used to index resources at BGS. Other preferred knowledge resources include a lexicon of rock units[3], gazetteers of place

[2]BGS Geoscience Thesaurus: https://www.bgs.ac.uk/discoverymetadata/13603129.html
[3]BGS Lexicon of Named Rock Units: http://data.bgs.ac.uk/doc/Lexicon.html

names, and a rock classification scheme[4]. There was less preference for more specialised knowledge-rich resources like Chemical analytes (selected by 5.6%) and Fossil taxonomy (selected by 11.1%) for use in information retrieval.

3.1.5 Summary of key survey findings

Findings from the survey responses are important for implementing search systems in general and semantic search tools in specific. Some findings are already confirmed by previous studies. For example, search application users benefit from promoting the relevant documents in search results. The ranking of search results is a major motivation for research in information retrieval. Ranking the most relevant documents highest in returned search results ensures that users do not spend time assessing irrelevant entries. Irrelevant entries in search results make it challenging to locate the relevant documents. The domination of irrelevant entries in the top ranks of search results exacerbates this problem.

Other findings that are more specific to using ontologies for search and they include the following.

1. It is preferable to present search application users with alternative meanings of ambiguous query terms to select intended meanings. Returned search results should reflect the disambiguation of query terms by promoting documents which express specified search intent. Achieving this requires an understanding of alternative meanings which can be realised through the semantic annotation of queries and documents.

2. Features for returning the results of narrower terms (or child terms), equivalent terms or alternative spellings of original search intent will benefit users. Use of these features should be optional with users having the ability to turn it on or off. Narrower terms are usually available in domain ontologies (e.g. thesauri, controlled vocabularies). At times, ontologies specify equivalent terms and spelling variations of concept labels as alternative textual labels of concepts. The respondents identified the Geoscience thesaurus and Lexicon of Named Rock Units as the most useful ontologies for this purpose.

[4]BGS Earth Material Classes: http://data.bgs.ac.uk/doc/EarthMaterialClass.html

3. The less specialised ontologies are used more regularly for search than the specialised ontologies. The less specialised ontologies are also lightweight ontologies, which makes them easier to generate and maintain. Multiple lightweight ontologies can be combined in a retrieval system and used as a unit through ontology alignment.

3.2 Datasets

In this section, we describe the datasets used for the evaluation of the contributions of this thesis.

3.2.1 Ontology alignment

The datasets for evaluating our ontology alignment methods are from the 2016 Ontology Alignment Evaluation Initiative (OAEI) Benchmark and Conference tracks [5]. The OAEI is an international initiative which coordinates the evaluation of alignment systems by providing a platform to compare them on different alignment problems/datasets. The features of the datasets are summarised in Table 3.1.

Table 3.1: Features of alignment datasets

Dataset	Ontology	Classes	Datatype properties	Object properties
Conference	Cmt	36	10	49
	ConfTool	38	23	13
	Edas	104	20	30
	Ekaw	74	0	33
	Iasted	140	3	38
	Sigkdd	49	11	17
	Sofsem	60	18	46
Benchmark	301	56	40	0
	302	24	25	5
	303	207	0	72
	304	114	11	40

The benchmark datasets are of the domain of bibliographic references, and each ontology is aligned to a reference ontology (test #101). We use the 4 ontologies specified as real-world ontologies (tests #301 to #304). The gold standards are reference alignments of

[5] http://oaei.ontologymatching.org/2016/

each ontology to test #101. The conference dataset consists of 7 small to medium-sized ontologies specifying concepts in the domain of conference organisation. The ontologies originated separately giving them highly heterogeneous characteristics. The gold standard is 21 reference alignments representing all alignments between pairs formed from the 7 ontologies.

Both object properties and datatype properties specify the values for entities. Object properties relate individuals to individuals (e.g. hasSibling) while datatype properties relate individuals to literals (e.g. year).

3.2.2 Semantic document annotation

Each dataset consists of an annotated corpus and one or more corresponding KRs. Table 3.2 summarises the features of the datasets in our evaluation.

Table 3.2: Comparison of the features of evaluation datasets

	Geology	**Computing**
No. of documents	397	195
Avg. document length	984.85 words	7,924.80 words
Content overlap	True (Nested document sections)	False (Entire documents)
Source of annotations	Directly assigned from vocabularies	Keywords selected from documents
No. of concepts	276 (used 701 times)	201 (used 416 times)
Annotations per doc.	1.8	2.1
Concept reuse	2.54	2.07

Geology: The first dataset was generated from 1,948 document sections in 30 geological memoirs. Domain experts manually annotated these document sections as part of a project aimed at enhancing content access. The memoirs are book-like documents with multiple sections[6]. Figure 3.7 is an example of a document section having two concepts from domain vocabularies as its annotation. The entries "value" and "scheme" refer to concepts and their source thesaurus respectively. We selected 3 controlled vocabularies that were used to annotate the documents – BGS Geoscience Thesaurus (GEOTHES)[7], BGS Geochronology (GEOCHRON)[8] and BGS Lexicon of Named Rock

[6] An example of geological documents used in evaluation http://pubs.bgs.ac.uk/publications.html?pubID=B01745

[7] http://www.bgs.ac.uk/discoverymetadata/13603129.html

[8] http://data.bgs.ac.uk/doc/Geochronology.html

Units (GEOLEX)[9]. GEOTHES and GEOLEX use broader and narrower relationships of the Simple Knowledge Organization System (SKOS) to specify their taxonomic structures. Chronostratigraphic subdivisions, which are informal "is-a" relationships, specify the taxonomic structure of GEOCHRON. Concepts from these vocabularies were used 701 times (276 unique concepts) to annotate 397 document sections making an average of 1.8 concepts per document section. We use these concepts (110 from GEOTHES, 122 from GEOLEX, and 44 from GEOCHRON) and the relevant subset of document sections for our evaluation.

```
<section>
    <sectioninfo>
        <indexterm scheme="CHRONOSTRAT" value="KU"> </indexterm>
        <indexterm scheme="AMF" value="5283"> </indexterm>
    </sectioninfo>
    <title>MID-CRETACEOUS TO END CRETACEOUS REGIONAL SHELF SUBSIDENCE</title>
    <para>Crustal extension had effectively ceased by mid-Cretaceous times (e.
    established over the Manx region, with structural demarcation between bloc
    Ireland implies deposition of Chalk across the entire Manx region. The max
    </para>
</section>
```

Figure 3.7: Example of annotated document section.

Computing: The second evaluation dataset has 244 scientific articles from SemEval 2010 Task 5[10]. Both authors and readers manually assigned document tags keywords with an average of 15.1 keywords per document. We use the ACM Computer Classification System (ACMCCS)[11] with 2,299 concepts as the KR for annotation. The KR does not contain all the document tags, so we use string comparison after stemming to identify instances where document tags match concept terms as in Große-Bölting et al. (2015). The outcome of this process is a set of tags for each document with corresponding entries in ACMCCS. ACMCCS uses SKOS broader/narrower relationships to specify its taxonomic structure. There were 416 concepts (201 unique) annotating 195 documents making an average of 2.1 concepts per document.

3.2.3 Semantic document retrieval

The evaluation of semantic document retrieval used datasets from TREC 2006 and 2007 Genomics tracks[12]. The 2016 and 2017 Genomics tracks have 28 and 36 queries/topics

[9] http://data.bgs.ac.uk/doc/Lexicon.html
[10] http://semeval2.fbk.eu/semeval2.php?location=tasks
[11] https://www.acm.org/publications/class-2012
[12] https://trec.nist.gov/data/genomics.html

respectively. Both tracks have a combined 62 queries after removing two queries with no relevant documents (topics 173 and 180).

The Genomics tracks use the same document collection consisting of 162,259 documents from 49 journals (12.3GB). We use an RDF version of Medical Subject Headings (MeSH) describing 23,885 concepts and a maximum taxonomic depth of 9 as domain ontology for semantic ranking[13]. MeSH is a controlled vocabulary which is used to index resources in the biomedical domain, and it aligns with our evaluation dataset. This dataset is suitable for our evaluation because it contains both documents and ontology of the same domain.

3.3 Evaluation Metrics

Evaluation metrics are appropriately chosen to provide insights into performances and comparative analysis of different approaches. We present details of metrics used to evaluate semantic document annotation, ranking in semantic document retrieval, and the alignment of ontologies. We use standard precision, recall and F1-measures for evaluation since mentioned areas all relate to information retrieval. The rest of this section describes these performance metrics for the experiments discussed in this thesis.

3.3.1 Evaluation of ontology alignment

In ontology alignment, an alignment system returns a set of alignment correspondences between a source ontology and a target ontology. Alignment performance is determined by comparing returned alignment correspondences with a reference alignment. The reference alignment contains the most accurate set of alignment correspondences for the ontologies and is usually generated manually by domain experts. A correspondence returned by a system is correct only if it is present in the reference alignment. Accordingly, we describe the precision (P), recall (R) and F1-measure (F) of alignment performance as follows:

$$\mathrm{P} = \frac{|\{\text{correspondences returned}\} \cap \{\text{correspondences in gold standard}\}|}{|\{\text{correspondences returned}\}|} \tag{3.1}$$

[13] https://old.datahub.io/dataset/mesh_ipsv_skos_rdf

$$\mathrm{R} = \frac{|\{\text{correspondences returned}\} \cap \{\text{correspondences in gold standard}\}|}{|\{\text{correspondences in gold standard}\}|} \tag{3.2}$$

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{3.3}$$

3.3.2 Evaluation of semantic annotation

Semantic annotation systems are evaluated using precision and recall measures. An annotation system returns a set of concepts from KRs with which to annotate a document. The annotation precision is the proportion of returned concepts that is correct annotations as specified by a gold standard. The annotation recall is the proportion of the correct concepts (specified by the gold standard) that is returned by an annotation system. F1 measure is the harmonic mean of precision and recall.

Let u represent the concepts which annotate a document in the gold standard and v represent the concepts that were selected to annotate the document by an annotation approach. Annotation performances are measured as follows as shown in equations 3.4, 3.5 and 3.3 for precision, recall and F1 measures respectively.

$$P = \frac{1}{|D|} \cdot \sum_{i=1}^{|D|} \frac{|u_i \cap v_i|}{|v_i|} \tag{3.4}$$

$$R = \frac{1}{|D|} \cdot \sum_{i=1}^{|D|} \frac{|u_i \cap v_i|}{|u_i|} \tag{3.5}$$

As shown in equation 3.6, Mean Average Precision (MAP) combines the precision and ranking quality of returned annotations in a single performance measure making it easier to compare different systems.

$$MAP = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{1}{\min(m, n)} \sum_{k=1}^{n} (p(k) \cdot rel(k)) \tag{3.6}$$

$p(k)$ is the precision at rank k of rank-ordered concepts selected, $rel(k)$ is 1 if the selected concept at rank k is correct (i.e. included in the gold standard) and 0 otherwise. m is the

number of relevant concepts returned as the annotation of d, n is the maximum number of returned concepts evaluated for each annotation approach, and D is the document collection being annotated.

3.3.3 Evaluation of retrieval results

We use the Mean Average Precision (MAP) to measure performances when evaluating the ranking of documents by retrieval systems. MAP combines precision, recall and ranking quality in a single performance measure making it easier to compare multiple systems. Accordingly, we determine the MAP of retrieval systems as shown in equation 3.7.

$$\text{MAP} = \frac{\sum_{q=1}^{|Q|} \text{AP(q)@n}}{|Q|} \tag{3.7}$$

Q is the set of all queries, $AP(q)@n = \sum_{k=1}^{n} P(k)/\min(m,n)$ is the average precision (AP) of retrieval for query q, $P(k)$ is the precision at position k, m is the number of relevant documents for the query, n is the maximum number of ranked search results being evaluated.

Another commonly used metric for evaluating ranking performance is the Normalized Discounted Cumulative Gain (NDCG). NDCG measures relevance quality using a graded scale and penalises the occurrence of a highly relevant document in the lower ranks of the search result entries by reducing its relevance logarithmically proportional to its position (Järvelin and Kekäläinen, 2002). The main difference between NDCG and MAP is that while MAP assumes a binary decision on relevance (that is, either relevant or not relevant), NDCG allows relevance to be specified as real number values. We use the MAP for evaluation given the nature of the evaluation dataset because it did not indicate relevance as a graded scale with which to order documents.

3.4 Chapter Summary

In this chapter, we presented the findings of a questionnaire survey on attitude to search and reception of semantic search features using ontologies of the geoscience domain. Use of ontology-based search features is popular in domains such as the biomedical domain, and the survey findings indicate that other domains can benefit from semantic search.

Also, we discussed the features of the datasets used in this research for evaluating proposed methods of ontology alignment, semantic annotation, and ontology-driven document search. Finally, we outlined the criteria for evaluating performances and comparing the results of different systems in this thesis.

Chapter 4

Semantic Ontology Alignment

Specific applications usually drive the creation of ontologies, and as a result, most ontologies cover specific sub-areas of domains. In order to achieve broader domain coverage or to cover overlapping domains, the identification of relationships between the concepts of different ontologies becomes essential. Establishing links across ontologies enables cross-ontology reasoning and the ability to merge multiple ontologies. As a result, a single ontology can be substituted by multiple ontologies in an application. Ontology alignment or ontology matching deals with the discovery of semantic correspondences between the entities of ontologies of overlapping domains. Ontology alignment techniques are crucial for integrating heterogeneous data sources and forms a crucial component for realising the vision of a Semantic Web.

In this chapter, we discuss the ontology alignment process and introduce two novel ontology alignment approaches which integrate string-based similarity and semantic similarity features using word embedding. The first approach is a supervised approach based on generating a machine classifier model using a hybrid of element-level string-based features, semantic similarity features, and context-based structure-level features. The second are unsupervised hybrid similarity models for element-level matching. They integrate string-based similarity, semantic similarity and term weight in a single similarity metric.

The supervised approach assumes the existence of validated alignment correspondences in a similar domain as the ontologies being aligned (training data). A machine learning

algorithm is trained on how to identify alignments using the validated alignment correspondences. The motivation for the unsupervised alignment methods is for alignment scenarios with no training data (or insufficient training data) for a supervised approach. However, unsupervised approaches often have to use multiple alignment techniques which presents the challenge of how to combine them as discussed in section 2.7.2.

In the rest of this chapter, we formalise the alignment problem and introduce the supervised and unsupervised alignment approaches. Afterwards, we evaluate the alignment approaches on benchmark datasets and discuss findings from the results.

4.1 Problem Definition

The ontology alignment process is challenging, especially when the ontologies are of different origins leading to inherent differences between them. Ontologies can vary vastly in levels of specification and vocabulary use even when they are from similar domain. The predominant methods for alignment use a composition of multiple string-based similarity metrics on textual features of entities (Cheatham and Hitzler, 2013). Semantic matching, rather than purely string-based matching, is essential for discovering correspondences by meaning when the vocabularies of the source and target ontologies differ. However, there is a shortage of semantic matching techniques (Otero-Cerdeira et al., 2015; Shvaiko and Euzenat, 2013). Semantic matching approaches often rely on the use of lexical databases such as WordNet which lack sufficient coverage. The lack of sufficient coverage in lexical databases becomes pronounced when dealing with domain-specific terminologies. As a result, word embedding approaches which are useful for capturing language semantics, have been proposed for semantic matching in ontology alignment (Sun et al., 2017; Zhang et al., 2014). Semantic matching approaches do not always outperform string-based similarity and effectively combining both strategies in alignment systems remains a challenge (Otero-Cerdeira et al., 2015).

An ontology, θ specifies a set of concepts (or entities), $\theta = \{c_1, ..., c_n\}$. A concept $c \in \theta$ represents the semantic definition of a meaningful entity in a domain. Although some ontologies also specify data properties and object properties, we use this minimal specification to include knowledge-light ontological resources such as thesauri and controlled vocabularies. Let $l(c)$ be the set of textual labels of a concept including alternative names (or synonyms), $tok(l_i)$ be the individual words of a concept's label, and $l(\theta)$ be an ontology's document collection which is all the labels of all concepts of θ. Let D and

D' be the document collections for θ and θ' respectively. To illustrate with Figure 4.1, concept #3945 has two labels making $l(\#3945) = \{$"petroleum industry","oil industry"$\}$, $tok($"petroleum industry"$) = \{$"petroleum","industry"$\}$, and $l(\theta)$ returns eight textual labels representing the documents in ontology collection D. We assume that the ontologies being aligned specify some form of subsumption relations between concepts such as "is-a" or "broader-than" relations. This allows for the identification of a concept's semantic context and depth in the ontology structure. The subsumption relation between two concepts c_i and c_j is represented as $c_i \prec c_j$ specifying that c_i is a broader concept of c_j (e.g. $\#2673 \prec \#3945$ in Figure 4.1).

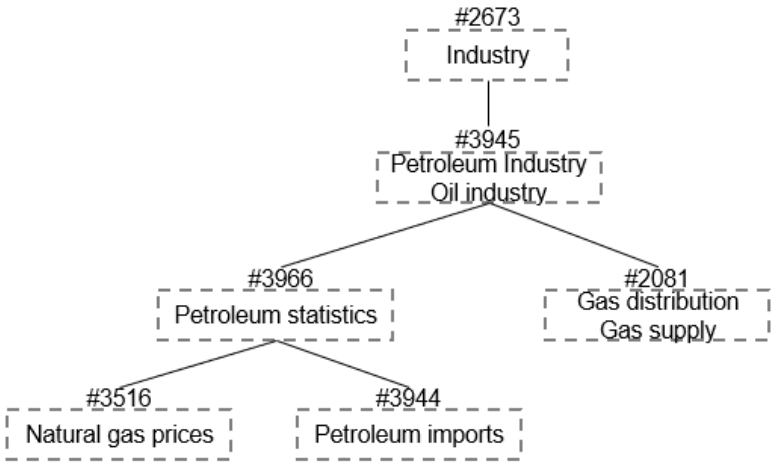


Figure 4.1: Example of concepts' hierarchy from a geoscience thesaurus with textual labels shown.

The output of the alignment process between the source ontology θ and target ontology θ' is the alignment, A which is a set of correspondences between semantically equivalent concepts of both ontologies. Each correspondence $a \in A$ is a 4-tuple, $a :< c, c', \equiv, \alpha >$ where $c \in \theta$, $c' \in \theta'$, $\equiv$ indicates equivalent relation type between c and c', and α is the confidence of alignment correspondence in $[0.0, 1.0]$ interval. Confidence is either 1 (correspondence) or 0 (no correspondence) for crisp alignment [1]. Given a source and target ontology, the problem is that of determining their alignment.

[1]Standard alignment with a clear demarcation between aligned and not aligned as opposed to having degrees of alignment (e.g. percentage or probability of alignment).

4.2 Supervised Ontology Alignment

Figure 4.2 presents a high-level overview of the ontology alignment process highlighting the training and testing/predicting phases. The supervised ontology alignment approach uses a composition of multiple similarity measures in a machine learning setting. The alignment process starts with the selection of candidate alignments using a variety of basic matching techniques. We then generate a feature vector composing of string-based similarity features and semantic similarity features for each correspondence in the candidate alignments. In the training phase, generated feature vectors are used to train a classifier. A pre-existing reference alignment specifies the correspondences between ontologies. The reference alignment is usually manually created and validated. The comparison between candidate alignments and the reference alignment results in a binary class of true correspondences and false correspondences. An alignment correspondence is a true correspondence when it is present in the reference alignment and a false correspondence when absent from the reference alignment. In the testing phase, we pass feature vectors to the machine classifier which determines whether each pair of concepts are true correspondence or false correspondence. The rest of this section discusses the stages of the ontology alignment process in detail.

4.2.1 Selection of Candidate Alignments

The alignment process begins with the identification of a set of candidate correspondences between the ontologies by comparing each concept from the source ontology with all target ontology's concepts. The objective of selecting candidate alignments is to avoid including concept pairs that have little or no chance of being aligned in the subsequent alignment filtering stage using a machine classifier. The selection of candidate alignments also avoids having to generate feature vectors for concept pairs with very low similarities and also leads to a better class balance when building a model. A pair of concepts become candidate alignments if their similarity exceeds the threshold for any of four similarity measures. The similarity thresholds for candidate selection are kept low to maximise recall. We also use a $Max1$ selection approach for each similarity measure such that if multiple concepts in the target ontology exceed the selection threshold when paired with a concept in the source ontology, we only choose the concept pair with the highest similarity value. $Max1$ is commonly used to enforce a one-to-one correspondence in alignments (Shvaiko and Euzenat, 2013). Accordingly, the cardinality of the set of

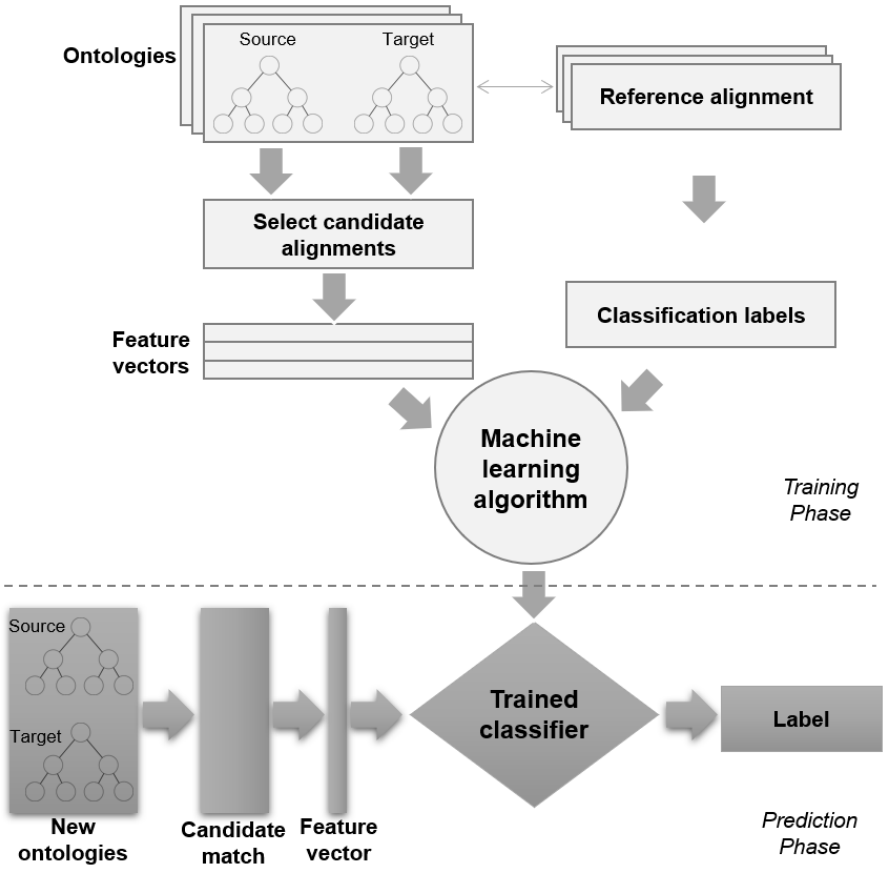


Figure 4.2: Overview of supervised ontology alignment process showing the training and prediction/testing phases.

candidate alignments is in the range $(0, 4 * min(|\theta|, |\theta'|)$ instead of the entire similarity matrix. The similarity measures were chosen after analysing a variety of ways in which concepts can be similar as follows.

1. Hybrid similarity (*hybrid*): combines similarity of word embedding vectors and similarity based on edit distance (Levenshtein).

2. Vector space model (*vsm*): cosine similarity of term vectors using term frequency-inverse document frequency (TF-IDF) scheme.

3. String metric for ontology alignment (*isub*): string similarity metric specially designed for ontology alignment.

4. Context similarity (*context*): indirectly compares concepts based on the hybrid

similarity between their semantic contexts (parent and child nodes).

Hybrid similarity

Hybrid similarity combines the use of semantic similarity using word embedding vectors and string-based similarity using edit distance measures. This similarity approach is expected to produce results that are at least as good as its components for element-level matching (Zhang et al., 2014). We use the word2vec skip-gram architecture to generate word embedding vectors (Mikolov et al., 2013a). The skip-gram model embeds the words from a corpus in a dense continuous vector space such that it generates similar representations for words that share similar contexts. For a pair of terms t_1 and t'_2 appearing in the word embedding vocabulary, $vecSim(t_1, t'_2)$ is the cosine similarity between the vector representations $\vec{t_1}$ and $\vec{t_2}$ respectively. String-based similarity using edit distance allows for fuzzy matching of strings. String matching is especially useful when comparing words that do not appear in the vocabulary of the word embedding model or variants of words due to inflexion (e.g. "apple" vs "apples"). The string-based similarity component of hybrid similarity uses the Levenshtein distance. The Levenshtein distance between two strings is the minimum number of character edits required to convert one string into the other. In contrast to Zhang et al. (2014), we impose a threshold on the string similarity component. The introduction of a threshold is because similarity due to common characters is no more than a coincidence at low similarity values. Measuring the hybrid similarity between terms follows the approach for measuring sentence similarity (Li et al., 2006). The best similarity coupling between the individual words of both strings determines their similarity as shown in equation 4.1.

$$hybrid(c, c') = \max_{\{l \in l(c), l' \in l(c')\}} \left\{ \frac{1}{maxLen(l, l')} \sum_{w \in l} max(emb(w, w'), lev(w, w')) \right\} \quad (4.1)$$

$maxLen(l, l') = max(|tok(l)|, |tok(l')|)$ is the length of the longer label, $emb(w, w')$ is the cosine similarity between the word embedding for w and w', and $lev(w, w')$ is normalised Levenshtein similarity.

Normalised Levenshtein similarity is calculated by first normalising the Levenshtein distance to a $[0.0, 1.0]$ range by diving by the length of the longer string. Similarity is then determined as $1 -$ normalised distance as in equation 4.2.

$$lev(w, w') = \rho \cdot (1 - \frac{Levenshtein(w, w')}{max(|w|, |w'|)}) \tag{4.2}$$

$$\rho = \begin{cases} 1, & \text{if } lev(w, w') \geq \beta \\ 0, & \text{otherwise} \end{cases}$$

β is the threshold for string-based similarity which is empirically determined from the training data. A threshold of about 0.88 is expected to be appropriate based on the results of edit distance approaches at the OAEI challenges. String similarity only contributes towards the final similarity of terms when it is up to the similarity threshold as determined by ρ.

In other words, equation 4.1 compares each and every word from one label with each and every word in the label being compared, and selects the maximum similarity of either word embedding or edit distance. The sum of best pairwise similarities is then divided by the length of the longer label. For example (and still using Figure 4.1), in comparing "oil industry" and "petroleum industry", the best similarities are $emb(\text{oil}, \text{petroleum}) = 0.65$ using the Google New word embedding model and $lev(\text{industry}, \text{industry}) = 1.0$ giving an overall similarity $\frac{1}{2}(0.65+1) = 0.825$. The most similar labels are used when concepts have multiple labels.

Vector space model (VSM)

The second similarity measure uses the classic vector space model. The cosine similarity of TF-IDF weight vectors determines the similarity between terms. Each ontology forms a collection, D with documents generated from labels of every concept ($D = labels(\theta)$). The TF-IDF weight for each word, w in a document, d (a concept's label) is determined as shown in equation 4.3.

$$tfidf(w, D) = f_{w,d} \cdot \log \frac{|D|}{n_w} \tag{4.3}$$

$f_{w,d}$ is the frequency of w in d, and n_w is the number of documents in which w appears. Using term weighting, frequently occurring words are made to contribute less to the final similarity of terms. The use of term weighting enables the discovery of alignments that

will otherwise be missed (Ngo et al., 2013). Cosine similarity between any two documents d and d' is then measured using TF-IDF weight vectors ($\vec{d}$ and $\vec{d'}$ respectively) as in equation 4.4.

$$\text{cosSim}(d, d') = \frac{\vec{d} \cdot \vec{d'}}{\|\vec{d}\| \|\vec{d'}\|} \tag{4.4}$$

The maximum similarity between the documents of concepts determines their VSM similarity since multiple documents can belong to a concept (see equation 4.5).

$$\text{vsm}(c, c') = \max_{\{d \in c, d' \in c'\}} \{cosSim(d, d')\} \tag{4.5}$$

The use of maximum similarity instead of other measures such as average similarity maximises recall which is one of the objectives of the candidate alignment stage.

String metric for ontology alignment

The third similarity approach is a string similarity metric, ISUB, specifically designed to align ontologies (Stoilos et al., 2005). The extent of common substrings is offset by their differences to determine the similarity between two strings as shown in equation 4.6.

$$\text{isub}(c, c') = \max_{\{l \in l(c), l' \in l(c')\}} \{Comm(l, l') - Diff(l, l') + winkler(l, l')\} \tag{4.6}$$

$Comm(l, l')$ is a function of common substrings, $Diff(l, l')$ is a function of the difference between the strings, and $winkler(l, l')$ is for improving the results. The Alignment API for ontology alignment includes an implementation of ISUB similarity (David et al., 2011).

Context similarity

When the lexical forms of textual features of a pair of concepts are different, comparing their ontological neighbourhoods can discover correspondences which are missed by direct comparisons. Accordingly, we indirectly compare concepts by the similarity of their semantic contexts. If the parents and children of the source and target concepts are

similar, we include the concepts in the set of candidate alignments. Let the immediate parent concepts of c be $P(c)$ and its immediate child concepts be $C(c)$, we implemented context similarity using the hybrid function defined above, as in equation 4.7.

$$\text{context}(c, c') = max \left\{ \frac{1}{2}(hybrid(c_p, c'_p) + hybrid(c_c, c'_c)) \right\} \tag{4.7}$$

max indicates that only the most similar parent and child concepts are used to determine context similarity with $c_p \prec c | c_p \in P(c)$, $c \prec c_c | c_c \in C(c)$, $c'_p \prec c' | c'_p \in P(c')$ and $c' \prec c'_c | c'_c \in C(c')$.

4.2.2 Concept Features for Alignment

In the second stage of the alignment process, we generate feature vectors for candidate alignments. A machine classifier uses the feature vectors to determine whether the correspondences in the candidate alignments are true correspondences. We introduce various novel features which we use in addition to commonly used similarity metrics for element-level concept matching. Features are grouped into three categories (selection, direct similarity, and context features) and summarised in Table 4.1. Recall that each candidate alignment correspondence comprises of a concept from the source ontology ($c \in \theta$) and the most similar concept to it in the target ontology ($c' \in \theta'$). We also note the next most similar concept to c in the target ontology ($c'' \in \theta'$) to determine features which are related to similarity offsets. All numerical features are generated to be in the [0,1] interval for ease of comparison and use by machine learning algorithms without requiring further normalisation.

Selection features

These features are determined during the selection of candidates alignments to reflect the best similarity value (sim), the method of similarity used ($matchType$), and similarity offset to the next most similar concept in target ontology ($simOffset$). $matchType$ is a nominal attribute used to indicate the similarity method that was used to select a candidate alignment. sim is determined as $max(hybrid(c, c'), vsm(c, c'), isub(c, c'), context(c, c'))$. $simOffset$ is determined as

Table 4.1: Feature vectors for alignment

Feature category	Feature	Description
Selection	$matchType$	Similarity method to select alignment
	sim	$max(hybrid, vsm, isub, context)$
	$simOffset$	Offset to the next best sim
	$hybrid$	Combines lev and emb
	vsm	Similarity based on vector space model
	$isub$	String similarity for ontology alignment
	$context$	$hybrid$ of semantic contexts
Direct similarity	lev	Similarity based on Levenshtein distance
	$fuzzy$	Fuzzy string score gives bonus points as characters in matched substrings increases.
	lcs	Similarity based on Longest Common Subsequence
	$dice$	Similarity based on Sorensen-Dice coefficient
	$mongeElkan$	Monge-Elkan similarity measure
	$prefixOverlap$	Prefix overlap divided by length of shorter string
	$suffixOverlap$	Suffix overlap divided by length of shorter string
	emb	Similarity of word embedding vectors
Context	$parentsOverlap$	Hybrid similarity of parent concepts
	$childrenOverlap$	Hybrid similarity of child concepts
	$contextOverlap$	Hybrid similarity of all context words
	$contextOverlapOffset$	Offset to next best $contextOverlap$
	$hasParents$	Indicates if both, one, or none of the concepts have parent nodes
	$hasChildren$	Indicates if both, one, or none of the concepts have child nodes
	$depthDiff$	Difference in relative depths of concepts

$sim(c, c') - sim(c, c'')$ and this captures how distinct the similarity of c and c' is, compared to the similarity of c and any other concept in the target ontology, θ'. High sim and $simOffset$ values are expected to be good indicators of a true correspondence. Finally, we also include each of the similarity methods used to select the candidate alignments

as a separate feature.

Direct similarity features

This category comprises other similarity metrics that directly compare textual labels of concepts. These include five commonly used string-based similarity measures – Levenshtein (*lev*), Fuzzy Score[2] ($fuzzy$), Longest Common Subsequence (*lcs*), Sorensen-Dice (*dice*), and Monge-Elkan ($mongeElkan$) (Cheatham and Hitzler, 2013; Monge et al., 1996). These were chosen to provide a variety of string similarities as each algorithm differs in its approach. Also, we include features for similarity based on word embedding alone (*emb*) and maximum prefix overlap ($prefixOverlap$) and suffix overlap ($suffixOverlap$) of concept labels. Prefix overlap and suffix overlap represent the numbers of contiguous characters shared at the beginning and end of strings respectively and are normalised by dividing by the length of the shorter string. Most of the string similarity measures were implemented using a publicly available API[3].

Context features

The placement of concepts on the ontology structure determines the features in this category. These include *parentsOverlap* and *childrenOverlap* which are *hybrid* similarities of parent and child concepts (of candidate nodes) respectively. We also introduce *contextOverlap* which is the *hybrid* similarity between all context words. That is, $contextOverlap(c, c') = hybrid((P(c) \cup C(c)), (P(c') \cup C(c')))$. $contextOverlapOffset$ is given as $contextOverlap(c, c') - contextOverlap(c, c'')$. Furthermore, we introduce two features ($hasParents$ and $hasChildren$) for additional insight into the neighbourhood of candidate alignments. $hasParents$ uses nominal features to indicate whether both concepts in a candidate alignment have parent nodes, only one concept has parent nodes, or neither have parent nodes. Similarly, with $hasChildren$, we indicate the presence or absence of child nodes. Finally, $depthDiff$ represents the absolute difference between the relative depths of the source and target concepts. The depth of a concept is the number of edges in the shortest path between the root node and that concept. We assume the presence of a top concept (root node) even when an ontology does not specify one. A concept's relative depth is the ratio of its depth to the total number of edges on the

[2] https://commons.apache.org/proper/commons-text/apidocs/org/apache/commons/text/similarity/FuzzyScore.html

[3] http://github.com/tdebatty/java-string-similarity

concept's path (i.e. from root to leaf passing through the concept). In Figure 4.1 for example, the relative depth of concept #3945 is 0.5 since #3945 is halfway down on the shortest path. We use the shortest path distances because they are not affected by the degree of interconnectedness in the ontology. This way, we get a fairer comparison of depths for ontologies of different structures.

4.2.3 Classification of Candidate Alignments

The final step is the classification of candidate alignments as either true or false correspondences. We use a Random Forest classifier which is an ensemble method using multiple decision trees for improved classification. Each decision tree in the Random Forest uses a subset of features and classification is based on majority voting of decision trees' predictions. Decision trees have been previously shown to outperform other machine learning algorithms for aligning ontologies (Ngo and Bellahsene, 2012). Our motivation for using the Random Forest classifier is due to its added advantage of avoiding overfitting when compared to a single decision tree (Breiman, 2001).

In the training phase of the classification process, we build a classifier model using ontologies whose alignment correspondences are known. Candidate alignments are selected and feature vectors generated for ontologies as summarised in Table 4.1. The reference alignments, which is the validated alignment correspondence between the ontologies, determine the class labels for training the classifier. When a correspondence from the candidate alignments is also present in the reference alignment, we label the correspondence as a true alignment correspondence. Otherwise, it is a false correspondence.

In the prediction (or classification) phase, the trained classifier model is used to determine if unseen alignment correspondences are true correspondences. The source and target ontologies go through the candidate selection and feature generation stages before applying the classifier. The ontologies in the training phase are expected to have features that are similar to the ontologies to be aligned in the prediction phase.

4.3 Unsupervised Ontology Alignment

Our supervised ontology alignment method assumes the existence of reference alignments with which it builds a classifier model. In the absence of reference alignments, or the

resources to create them, an unsupervised approach must be taken. In this section, we introduce two element-level matchers for unsupervised ontology alignment. Element-level matchers discover alignment correspondences by analysing concepts in isolation (Faria et al., 2013).

Our element-level matchers integrate edit distance for string-based similarity, word embedding vectors for semantic similarity, and a term weighting scheme to determine the relative importance of words in concept terms. Term weighting uses weighting factors to express the relative importance of different components of a whole. Term weighting is employed because we do not expect all the words of phrasal concept terms to be of equal importance. For example, when comparing concept terms "igneous" and "metamorphic rock" from ontologies of the geoscience domain, "metamorphic" is expected to weigh more than "rock" in the latter concept term. The inclusion or omission of qualifier terms such as "rock" can be due to the naming conventions of the different ontologies. Also, in order to maintain a high precision while improving recall, we limit the use of string-based similarity such that it does not contribute to overall similarity when it is below an empirically determined threshold.

4.3.1 Element-level Ontology Alignment

Figure 4.3 is a high-level overview of the alignment process. The pre-processing stage normalises concept labels by separating camel-cased and underscored strings to form individual tokens. We remove punctuations and stop-words before indexing concept terms.

The discovery of correspondences between the ontologies follows the usual approach of comparing each and every concept of source ontology to each and every concepts of the target ontology as shown in lines 2 to 13 of Algorithm 1. The target ontology concept with the highest similarity is selected as a correspondence if it exceeds a threshold. Empirically determined hyper-parameters, α and β, are similarity thresholds for overall similarity (ws) and string-based similarity (lev) respectively. Since a concept can have multiple labels (e.g. alternative entry names, synonyms), we evaluate each label of a source concept independently when measuring similarity with the concepts of the target ontology. The similarity for a pair of concept labels is the maximum similarity obtained from the pairwise comparison of all their labels. Line 5 of Algorithm 1 gives the overall similarity measure, and its implementation varies between the alignment approaches.

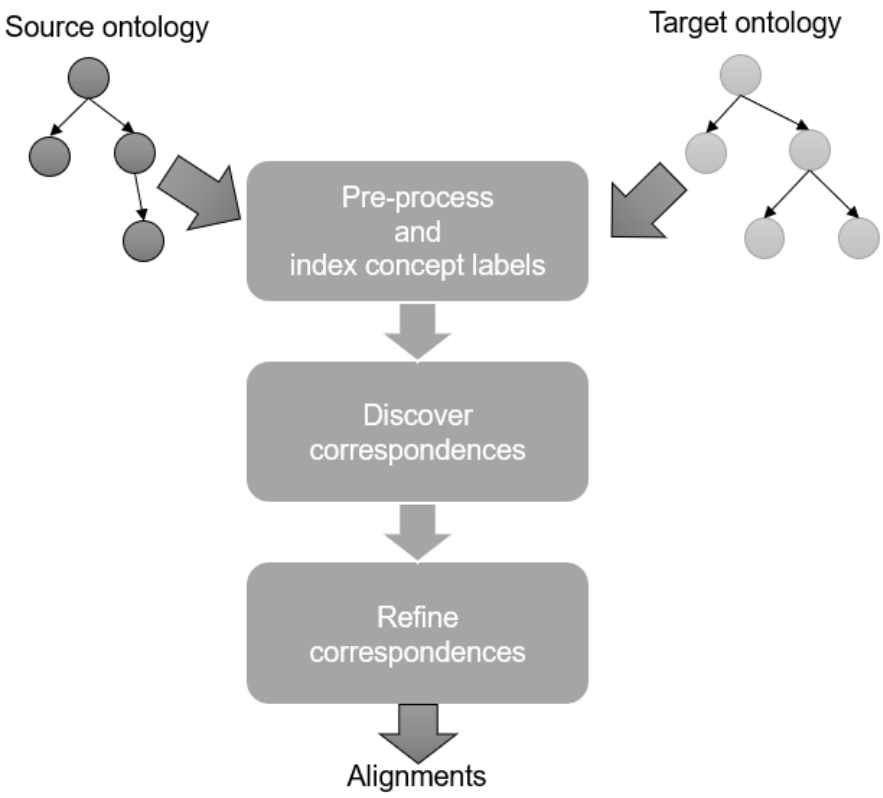


Figure 4.3: Pipeline for unsupervised alignment

We will discuss the determination of similarities in sections 4.3.2 and 4.3.3 for alignment approaches Weighted Hybrid Similarity and Weighted Vector Similarity respectively.

The refinement stage enforces a $Max1$ selection for each concept in the discovered alignment correspondences. To illustrate $Max1$ selection, let us assume that the correspondence, $a \rightarrow b$ has a similarity that is above the similarity selection threshold. A later comparison $c \rightarrow b$ with higher similarity supersedes the earlier correspondence. This approach is commonly used to enforce a one-to-one correspondence in final alignments (Shvaiko and Euzenat, 2013).

Term Weighting

Term frequency-inverse document frequency (TF-IDF) weight for term, t in a document $d \in D$ is determined as in equation 4.3. TF-IDF weights are normalised by dividing each coordinate by the document's length vector to obtain a weight as follows (equation 4.8).

Algorithm 1: Pairwise concept comparison for alignment

Input: θ, θ', α, β.
Output: *alignments*.

```
1  alignments ← {};
2  foreach c ∈ θ do
3  |  maxSim = 0, correspondence = null;
4  |  foreach c' ∈ θ' do
5  |  |
   |  |                    sim ← max_{s∈l(c),s'∈l(c')} ws(s, s')
   |  |
   |  |  if sim ≥ α  and  sim > maxSim then
6  |  |  |  maxSim = sim;
7  |  |  |  correspondence = (c, c', ≡, sim);
8  |  |  end
9  |  end
10 |  if correspondence ≠ null then
11 |  |  alignments ← alignments + correspondence;
12 |  end
13 end
14 return alignments
```

It is important to note that while weighting expresses relative important, normalisation rescales values to be within a specific range (the [0,1] interval in this case).

$$wt(t,d) = \frac{tfidf(t,d)}{\sqrt{\sum_{t' \in d} tfidf(t',d)^2}} \tag{4.8}$$

Recall that for our purpose, each concept label is a document and all concept labels in an ontology form a collection. We chose raw counts for term frequency since document sizes are short and comparable.

4.3.2 Weighted Hybrid Similarity

Our first matching technique is a weighted hybrid similarity (WHS) approach for element-level matching and combines semantic similarity, string-based similarity, and term weighting in a single similarity measure. Once again, we use the approach for measuring sentence similarities to compare strings (Li et al., 2006). We obtain the hybrid similarity between

a pair of words in Algorithm 1 as shown in equation 4.9.

$$ws(s, s') = \sum_{w \in s} wt(w, s) \cdot wt(w', s') \cdot N(w, s') \tag{4.9}$$

w and w' are tokens in s and s' respectively and $N(w, s') = h(w, w')$ is the normalising coefficient. $h(w, w')$ is the hybrid similarity of semantic and string-based similarity measures as follows.

$$h(w, w') = max\{emb(w, w'), lev(w, w')\} \tag{4.10}$$

$emb(w, w')$ is the cosine similarity between the embedding vectors of w and w', and $lev(w, w')$ is the normalised Levenshtein similarity as shown in equation 4.2.

The weighted hybrid similarity approach can be viewed as extensions of the hybrid element-level matching technique (Zhang et al., 2014) and soft term frequency-inverse document frequency (Soft TF-IDF) (Cohen et al., 2003).

4.3.3 Weighted Vector Similarity

The second element-level matching technique is a weighted vector similarity (WVS) approach that alters the magnitude of embedding vectors using TF-IDF weights. Weighted vectors are summed up for multi-word concept labels to obtain a new vector representation of the same dimensions as shown in equation 4.12. The effect of altering vector magnitudes before addition is that the direction of the new vector remains relatively close to the vector of words with the highest weights as illustrated by Figure 4.4.

In Figure 4.4, tokens t_1 and t_2 are words in concept label s having TF-IDF weights $wt(t_1, s)$ and $wt(t_2, s)$ respectively. By applying weight for $\vec{s_2}$, the new vector remains close to the more significant word. Accordingly, the weighted similarity is determined as follows.

$$ws(s, s') = max\{cosSim(\vec{s}, \vec{s'}), lev(s, s')\} \tag{4.11}$$

$cosSim(\vec{s}, \vec{s'})$ is the cosine similarity between vectors $\vec{s}$ and $\vec{s'}$) resulting from weighted

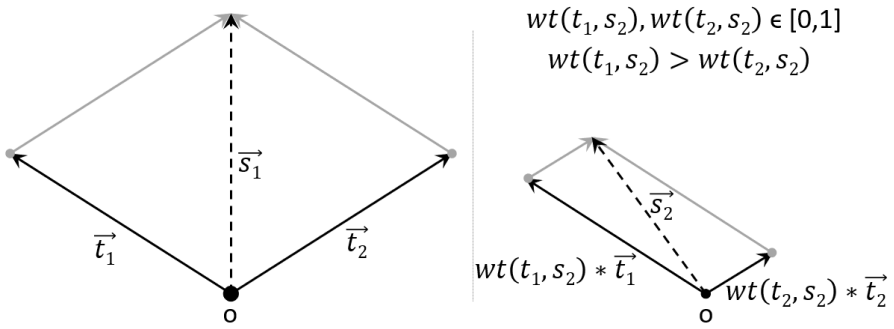


Figure 4.4: Unlike for $\vec{s_1}$, vectors are multiplied by TF-IDF weights prior to addition for $\vec{s_2}$. This disproportionately dampens the magnitude of vectors of less important words, thereby minimising their impact in subsequent vector addition.

vector additions. $\vec{s}$ is determined as shown in equation 4.12.

$$\vec{s} = \sum_{w \in s} \vec{w} \cdot wt(w, s) \tag{4.12}$$

We expect the weighted addition of vectors to be especially useful for discovering correspondences such as in "geological age" vs "geological dating". The term "geological" will weigh less if it has a high occurrence frequency in the ontologies thereby, reducing the magnitude of "geological" in the vector space. The reduced significance of "geological" places more emphasis on the vectors of "event" and "activity" for measuring similarity.

4.4 Evaluation

4.4.1 Datasets and experiment setup

We perform experiments to evaluate the performance of our alignment approaches using datasets from the Conference and Benchmark tracks of OAEI (see section 3.2.1 of Chapter 3). We use two word embedding models – a Wikipedia model and the Google New Negative300 model – as the word embedding models for determining semantic similarity. We generated the Wikipedia model using an open-source deep learning library[4] and a November 2016 database dump of Wikipedia English language articles. The Wikipedia model has 300 dimensions and omitted words which occurred less than 10 times in the

[4] https://deeplearning4j.org/word2vec.html

corpus. The Google New Negative300 model was generated by Google and is publicly available[5]. Using both models allows us to analyse the influence of switching word embedding models.

4.4.2 Alternative alignment approaches for comparison

- *StringEquiv*: A string matching approach that discovers alignments by comparing concept labels for exact string matches. *StringEquiv* is an OAEI baseline which outperforms several alignment systems.
- *edna*: An edit distance approach based on Levenshtein distance. *edna* is another OAEI baseline which uses edit distance for approximate string matching and outperforms *StringEquiv*.
- $ISUB$ is a string similarity metric specifically designed for ontology alignment (Stoilos et al., 2005).
- $WordNet$ measures similarity on WordNet using the Wu & Palmer edge-counting semantic relatedness algorithm (Blanchard et al., 2005).
- $WordEmb$ Word embedding approach using Google News Negative300 model.
- $Hybrid$ Combines word embedding and edit distance to discover correspondences (Zhang et al., 2014).

Our new supervised ontology alignment approach which we refer to as $Rafcom$ has two variants – $Rafcom_W$ and $Rafcom_G$ for Wikipedia-based and Google News word embedding models respectively. We use the leave-one-out cross-validation approach for the Conference dataset such that a pair of ontologies is left out in turn while we train a model with the remaining dataset. The trained model is then used to aligned the withheld ontologies.

We determine alignment performance using standard precision, recall and F1 measures averaged over the test data. Precision is the proportion of the returned set of correspondences that are present in the reference alignment. The recall is the proportion of correspondences in the reference alignment that are returned by an alignment system. F1-measure is the harmonic mean of precision and recall.

[5] https://s3.amazonaws.com/dl4j-distribution/GoogleNews-vectors-negative300.bin.gz

4.4.3 Results and discussion

Tables 4.2 and 4.3 show the performance results of the alignment approaches for Benchmark and Conference datasets respectively. Similar to OAEI challenges, we show results for the best F1-measure obtained for each approach. The best performance values for each evaluation metric are in boldface. The results show that the variants of $Rafcom$ outperformed other approaches on both datasets based on F1-measure, with $Rafcom_G$ slightly outperforming $Rafcom_W$. On the Conference dataset, the candidate selection stage discovered 84% of the true correspondences, and the classifier achieved 96% accuracy. On the Benchmark dataset, the selection of candidate correspondences discovered 82% of the true correspondences, and classification accuracy was 87%. Both $Rafcom_W$ and $Rafcom_G$ achieved high precision values while maintaining good recall values on both datasets. Figure 4.5 shows the results of alignment systems on the Conference dataset at the OAEI challenge ordered by F1-measure[6]. *edna* is a good baseline and outperformed $StringEquiv$ on both datasets which is consistent with results at the OAEI challenge and previous works (Cheatham and Hitzler, 2013). The results of $Rafcom$ are promising when compared with the best systems at the OAEI challenge. $Rafcom_W$ is slightly outperformed by $Rafcom_G$ indicating that the quality of the word embeddings in the Google New model were better suited for the alignment task.

Table 4.2: Results on OAEI 2016 benchmark track

	Precision	Recall	F1	Cut
$StringEquiv$	0.**96**	0.53	0.69	
$edna$	0.94	0.54	0.69	0.88
$ISUB$	**0.96**	0.54	0.69	0.96
$WordNet$	0.87	0.48	0.62	0.96
$WordEmb$	**0.96**	0.48	0.64	0.84
$Hybrid$	**0.96**	0.54	0.69	0.89
WHS	0.89	0.63	0.74	0.89
WVS	0.90	0.65	0.75	0.89
$Rafcom_W$	0.89	0.70	0.79	
$Rafcom_G$	0.89	**0.70**	**0.79**	

Unsupervised approaches WHS and WVS outperformed other element-level matchers on both datasets based on F1-measures. However, they were slightly below the performance of the supervised approaches. The weighted hybrid similarity and weighted vector similarity produced similar results as reflected in the similar performances of WHS and

[6] http://oaei.ontologymatching.org/2016/conference/eval.html

Table 4.3: Results on OAEI 2016 conference track

	Precision	Recall	F1	Cut
StringEquiv	0.88	0.50	0.64	
edna	0.88	0.54	0.67	0.88
ISUB	0.84	0.56	0.67	0.96
WordNet	0.85	0.56	0.67	0.95
WordEmb	0.88	0.56	0.68	0.84
Hybrid	0.88	0.58	0.70	0.85
WHS	0.79	**0.73**	0.76	0.76
WVS	0.77	**0.73**	0.75	0.77
$Rafcom_W$	0.89	0.68	0.77	
$Rafcom_G$	**0.90**	0.69	**0.78**	

Matcher	Threshold	Precision	F.5-measure	F1-measure	F2-measure	Recall
CroMatch	0	0.78	0.77	0.76	0.75	0.74
AML	0	0.83	0.8	0.76	0.72	0.7
LogMap	0	0.84	0.79	0.73	0.67	0.64
XMap	0	0.86	0.8	0.73	0.67	0.63
LogMapBio	0	0.8	0.76	0.71	0.67	0.64
DKPAOMLite	0	0.82	0.76	0.69	0.63	0.59
DKPAOM	0	0.82	0.76	0.69	0.63	0.59
edna	0	0.88	0.78	0.67	0.59	0.54
NAISC	0.98	0.85	0.77	0.67	0.59	0.55
FCAMap	0	0.75	0.72	0.67	0.63	0.61
LogMapLt	0	0.84	0.76	0.66	0.58	0.54
StringEquiv	0	0.88	0.76	0.64	0.55	0.5
Lily	0	0.59	0.6	0.61	0.62	0.63
LPHOM	0.76	0.89	0.71	0.55	0.45	0.4
Alin	0	0.89	0.65	0.46	0.36	0.31
LYAM	0.97	0.48	0.36	0.26	0.21	0.18

Figure 4.5: Performance of alignment systems on OAEI 2016 conference track (classes only).

WVS. *Hybrid* was better than its components (*edna* and *WordEmb*) as had been expected because string similarity can complement vector similarity in some comparisons and this agrees with the results in previous work Zhang et al. (2014). *WordEmb* performed slightly better than *WordNet* on both datasets. Although word embedding has a more extensive vocabulary, the concept terms in this evaluation were also found on WordNet as the domains are not very specialised. *WHS* and *Hybrid* share similar components, and the superior performance of *WHS* highlights the importance of term weighting and limiting the contribution of string-based similarity. Analysis of the performance of *WHS* reveals how different components contributed to the discovery of different types of correspondences that other unsupervised techniques failed to find. The

string-based similarity component allows spelling variations such as "Organisation" ≡ "Organization" to be correctly identified as a correspondence. In comparing "Conference Dinner" with "Conference Banquet", the similarity of word embeddings for "dinner" and "banquet" is 0.7 (well above 0.29 when comparing strings) which contributes to the correct selection of the pair as an alignment correspondence. Terms weighting in WHS helped to identify correspondences such as "Banquet" ≡ "Conference Banquet". "Conference" has a relatively high frequency in the ontology collection and therefore had a lower term weight than "banquet" in "Conference Banquet". The enhanced importance of "Banquet" led to an overall similarity that is above the selection threshold. Similarly for WVS, adding the weighted vector of "conference" to the vector of "banquet" only modifies it slightly such that it still yields a high similarity when compared with the vector of "banquet" alone. *edna* is a good baseline and was slightly better than *stringEquiv* which is consistent with results from the OAEI challenges.

Effect of hyper-parameters on unsupervised approaches

Hyper-parameters α and β influence the performances of WHS and WVS. A grid search was performed on the Conference dataset for an insight on the hyper-parameters affect performance. Figures 4.6 and 4.7 show the scatter boxplot of F1 measure as α and β are varied for WHS and WVS respectively.

α is best in the 0.7 to 0.8 region while β is best in the 0.8 to 0.9 region. Performances in these regions only differ slightly showing that the similarity models are not very sensitive to the variability of the hyper-parameters. The observed reduction in performance when we fix α and decrease β supports limiting the contribution of string metric to the overall similarity of concept labels. At lower thresholds, performance drops off more significantly for α than β indicating that string similarity threshold (β) had minimal positive contributions to overall performances. Therefore, the superior performances of WHS and WVS compared to $Hybrid$ are mostly due to the introduction of term weights.

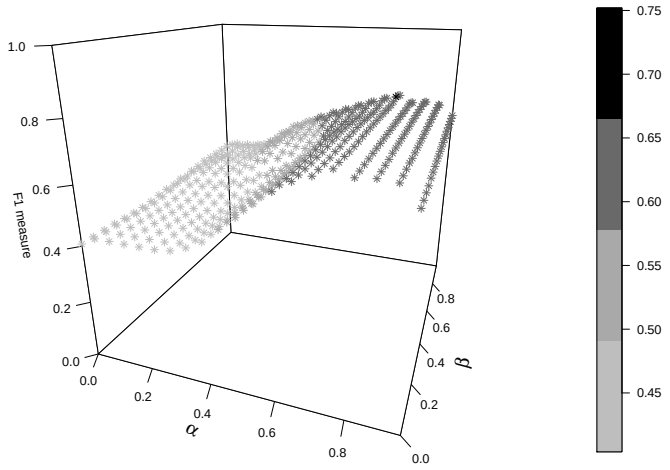


Figure 4.6: Influence of hyper-parameters on the performance of WHS

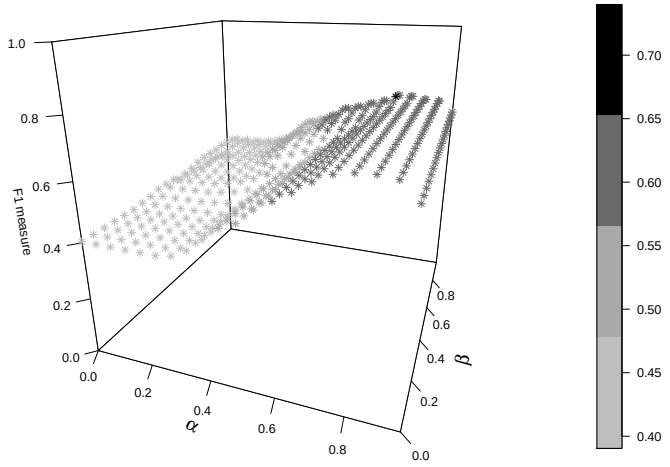


Figure 4.7: Influence of hyper-parameters on the performance of WVS

Similarity types in the discovery of candidate alignments

The discovery of alignment correspondences using exact string matches is the most straightforward alignment approach. Any of *hybrid*, *isub*, or *vsm* discovers such correspondences. There are observed differences between similarity approaches when concept labels do not match as shown in Table 4.4. The correspondence between "Academic_Event" and "Scientific_Event" was discovery using the Hybrid approach because

the word embedding model generates similar representations for “Academic” and “Scientific”. The correspondence between “Track-workshop_chair” and “Workshop_Chair” was discovered by ISUB similarity. ISUB similarity places more emphasis on common substrings than it offsets for differences resulting in a high similarity (0.91) for this example. In contrast, the similarity between this pair is 0.6 using Levenshtein. “Conference_document” and “Document” have a high similarity of 0.94 using VSM. The high VSM similarity is because “Conference” appeared multiple times in both ontologies (conference# and ekaw#) and as a result, has a low TF-IDF weight. “conference#Conference_document” vs “ekaw#Conference” results in a similarity of 0.33 using VSM highlighting the reduced significance of “Conference”. Also interesting is the comparison between “Paper” and “Submission” which returned low similarity scores for all direct comparisons. However, comparing their semantic neighbourhoods rightly identifies the pair as candidate alignments.

Table 4.4: Similarity values using different similarity measures for some correspondences discovered

Source concept vs Target concept	**Similarity approaches**			
	hybrid	**stoilos**	**vsm**	**context**
conference#Paper vs confOf#Paper	**1.0**	**1.0**	**1.0**	0.28
edas#Academic_Event vs ekaw#Scientific_Event	**0.84**	0.61	0.34	0.72
conference#Track-workshop_chair vs ekaw#Workshop_Chair	0.56	**0.91**	0.42	0.25
conference#Conference_document vs ekaw#Document	0.57	0.81	**0.94**	0.33
edas#Paper vs iasted#Submission	0.18	0.0	0.0	**0.76**

Influence of feature categories in supervised alignment

To evaluate how the features influenced performance, we perform experiments by dropping feature categories during classification of candidate alignments. As shown in Figure 4.8, precision and recall values were observed for each group of feature categories. We reused previous configurations and base performance on 10-fold cross-validation on the Conference dataset.

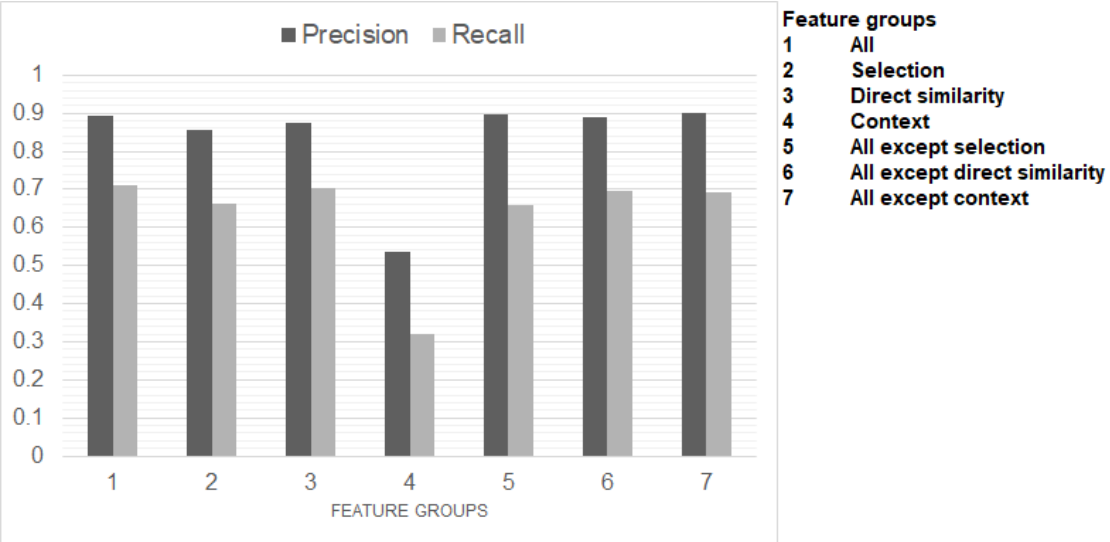


Figure 4.8: Impact of excluding features categories.

Classification using all features (1) was best but only marginally better than dropping the context features (7). Context features contributed least to the classification performance, and this is further highlighted by weak performance when we use context features alone for classification (4). We put this down to lack of sufficient data to learn to use context features. Analysis of candidate alignments showed that only 3% of the true correspondences were identified using context similarity. As a result, the classifier model did not learn to use context information effectively. An interesting observation is that using direct similarity features alone (3) can produce a good performance. However, dropping the direct comparison features (6) can produce a performance that is almost just as good. This contradictory observation suggests that several similarity features are redundant.

Distinctiveness of alignments discovered

In order to further investigate how the alignment correspondences returned by different matching techniques compare, we define an overlap function. Let X be the correspondences returned by matching technique a and Y be the correspondences returned by matching technique b. We express the overlap of a and b as shown in equation 4.13.

$$a \odot b = \frac{|X \cap Y|}{|X|} \tag{4.13}$$

Equation 4.13 gives a values in the $[0, 1]$ interval for any pair of alignment techniques so that $0 \leq a \odot b \leq 1$. $a \odot b = 1$ shows that all the correspondences returned by a were also returned by b while $a \odot b = 0$ show that there are no common correspondences between a and b. Overlap results for the conference dataset are as shown in Table 4.5. Row names represent a and column names represent b in the overlap comparisons.

Table 4.5: Overlap matrix of correspondences returned

$\odot$	*stringEquiv*	*edna*	*ISUB*	*WordNet*	*wordEmb*	*Hybrid*	*WHS*	*WVS*
stringEquiv	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
edna	0.93	1.00	1.00	0.97	0.97	1.00	1.00	1.00
ISUB	0.86	0.92	1.00	0.92	0.93	0.95	0.96	0.95
WordNet	0.89	0.93	0.95	1.00	0.97	0.96	0.96	0.96
wordEmb	0.89	0.93	0.96	0.96	1.0	1.00	0.99	0.99
Hybrid	0.86	0.93	0.96	0.94	0.97	1.00	0.99	0.99
WHS	0.61	0.66	0.68	0.66	0.68	0.70	1.00	0.97
WVS	0.61	0.66	0.68	0.66	0.68	0.70	0.94	1.00

Table 4.5 shows that WHS returned most of the correspondences that were discovered by the other techniques. On the other hand, none of the other techniques has a relatively high overlap with WHS. For example, 0.96 of the correspondences identified by $ISUB$ were also identified by WHS. In contrast, only 0.68 of correspondences identified by WHS were also returned by $ISUB$. $WHS \odot WVS = 0.97$ while $WVS \odot WHS = 0.94$ which highlights the difference in characteristics of both similarity models.

4.5 Chapter Summary

In this chapter, we introduced a classifier-based approach for ontology alignment based on a hybrid of string-based and semantic similarity features. Word embedding was used to generate semantic features for classification in addition to novel features which we introduced. Our experiments showed promising results and outperformed the previously known approach which incorporates word embedding. Also, comparison with the best-performing alignment systems at the OAEI challenge shows that it can outperform state-of-the-art systems.

Training data is not always available for supervised approaches. Accordingly, we introduced two hybrid similarity models for element-level matching as unsupervised ontology alignment techniques. The similarity models integrate term weights to a hybrid of word embedding-based semantic similarity and limited use of a string metric. The first model compares concept terms using individually weighted tokens while the second model generates a unified vector representation through weighted addition of vectors of component tokens. Experiments using OAEI datasets show that the similarity model outperforms other commonly used element-level matching techniques.

Both the supervised and unsupervised ontology alignment approaches introduced in this chapter rely on minimal information from the ontology to achieve competition alignment performances. The ability to use minimal ontology knowledge makes the alignment approaches suitable for aligning knowledge-light ontological resources such as taxonomies and controlled vocabularies. However, these alignment techniques can be improved further. Future work will investigate incorporating structure-level matching techniques in unsupervised approaches and improved post-alignment refinement using a reasoner.

Chapter 5

Semantic Document Annotation

This chapter addresses the conceptual representation of documents which is a requirement for concept-based information retrieval. Ontologies form an explicit semantic space between queries and target document collections in an ontology-based retrieval model. It is sufficient to link queries solely to concepts (or entities) when using ontologies for query expansion. When further integration of conceptual knowledge is required such as semantic ranking based on document entities, the need also arises to link documents to the ontological concepts. Semantic document annotation is the process of linking the content of documents to the concepts of ontologies that unambiguously describe them. Semantic annotation is an enabling technology which in addition to enriching content, enables applications for search and browse (Berlanga et al., 2014; Hulpus et al., 2013). We present our proposed method for semantically annotating documents which is based on the unsupervised generation of descriptive textual features for the concepts of an ontology, followed by a retrieval approach to identify the concepts which are most relevant to a document.

There are different levels of granularity in document annotation, and strategies for annotation differ accordingly. Levels of granularity include entire documents, sections or segments of a document or specific named entities in a document. While a high-level understanding of content may be sufficient when annotating an entire document, techniques such as named entity recognition, word sense disambiguation, and co-reference resolution are more pertinent to annotating specific terms or named entities. This chapter focuses on the annotation of entire documents or its segments (e.g. chapters and sections) which is especially useful for books and other publications that can cover a

range of domain topics. When accomplished, digital agents can reuse such annotations in bespoke ways such as for faceted search, content navigation and meeting information needs by dynamically assembling a pseudo-document from relevant segments of different documents. The strategies for annotating segments of documents can be generalised for annotating entire documents. Accordingly, we treat segments of documents as individual documents. Also, we use the terms thesaurus, controlled vocabulary and related terms to refer to ontological knowledge resources which specify domain concepts and have taxonomic structures.

State-of-the-art approaches to semantic document annotation rely on reusing concepts in similar annotated documents. This is intuitive since it assigns similar annotations to documents with similar contents. However, annotation reuse requires having an initial set of annotated content which poses two significant challenges. First is the challenge of generating the initial set of annotated documents. Second is the cold start problem as it is challenging to discover concepts that were sparingly used or not used in the initial annotated set. The challenge is even more pronounced when new concepts are introduced to the knowledge resource or existing concepts get modified. Most of the alternative approaches that do not require a pre-annotated corpus rely on thesaurus-based features (concept terms, synonyms, and descriptions) or specific mentions of concepts terms in documents. These can lead to poor results as controlled vocabularies can lack sufficient descriptive features for concepts with which to effectively link them to document contents.

In this chapter, we attempt to overcome the challenges of lack of a pre-annotated corpus or sparse descriptive concept features in a KR by generating an annotated set of pseudo-documents from an external corpus. Each generated pseudo-document represents a concept in the KR. We aim to ensure that concept summaries are sufficiently descriptive of the concepts of the KR such that unsupervised annotation approaches can use them to annotate new documents. Specifically, we address the following questions:

1. Can we generate descriptive textual features for the concepts of a taxonomic knowledge resource?

2. Can the generated textual features for concepts be used by unsupervised annotation approaches to annotate documents effectively?

5.1 Definition of Key Terms

Definition 1 *A knowledge resource (KR), θ formalises the semantics of a domain using a set of concepts (or entities), $\{(c_1, ..., c_n) \mid c_i \in \theta\}$.*

We assume the existence of a taxonomic structure in the KR. Determining the relationships between the concepts of a KR is enabled by a taxonomic structure. The taxonomy of concepts is specified using broader/narrower term relations or informal "is-a" relationships. We use the terms thesaurus and controlled vocabulary interchangeably, and in either case, we refer to a taxonomic KR that specifies domain concepts.

Definition 2 *A concept $c \in \theta$ represents the semantic definition of a meaningful entity in a domain.*

We assume that each concept has one or more textual natural language labels. The entirety of the textual features of a concept is its descriptive textual features which we refer to as its "concept summary". In other words, concept summary is a pseudo-document containing all the text about a concept. To illustrate this, consider a Wikipedia article to be a concept. The title of the article represents the concept label, the content of the Wikipedia article represents the concept summary, and other texts which redirect to the Wikipedia page are alternative concept labels (synonyms) of the concept. Although a KR can specify other properties for concepts such as data properties and object properties, we do not rely on these as they are often absent from thesauri and controlled vocabularies.

A key process in our approach is a novel method of determining the most relevant sources for concept summary generation from the external corpus. We use knowledge of semantic relatedness between the concepts of a thesaurus to identify the best documents from which to generate concept summaries. The taxonomy is used as a semantic filter to deal with synonymous and polysemous concept terms. Subsequently, we use the concept summaries for the unsupervised annotation of documents. The annotation process compares different representations of concept summaries and documents being annotated to identify the most suitable concepts for annotation. We also introduce an approach for measuring semantic precision and recall for evaluating annotation performance. We discuss the motivation and formalisation of the semantic evaluation approach.

The rest of this chapter is structured as follows: section 5.2 presents the generation of concept summaries, section 5.3 is the use of concept summaries for annotating documents, and section 5.5 is experimental evaluation of different approaches for annotating

documents using two contrasting datasets to highlight the strengths and weaknesses of each approach.

5.2 Corpus-based Concept Summaries

The motivation for generating concept summaries is to augment concepts with externally sourced textual features that effectively describe them. The disambiguation of terms in the external corpus for generating concept summaries uses the knowledge of concept relatedness. The entities in the semantic neighbourhood of a concept form a semantic filter for generating its summary. As a result, the semantic filtering strategy requires the existence of a taxonomic structure of concepts. A high-level overview of the process for generating a concept summary is presented in Figure 5.1. We summarise the process in the following steps:

1. The textual labels of a concept form a query to retrieve documents from a corpus. We refer to this concept as *query concept*.

2. Documents returned in step 1 are mapped to the thesaurus to identify the concepts expressed in them. We refer to the set of concepts mapped to a document as *document concepts*.

3. Each document in step 2 is re-ranked based on the semantic overlap between the query concept and document concepts. We use pairwise semantic relatedness measures between query concept and document concepts to estimate semantic overlap.

4. The concept summary is created by extraction-based summary generation using the top-ranked documents in step 3.

We repeat steps 1 to 4 for all the concepts of a thesaurus resulting in a corpus of concept summaries. The concept summaries are indexed for annotating new documents in the annotation phase. In the next section, we will present two approaches to using the concept summaries for annotation. The remainder of this section describes the above steps for generating concept summaries in more detail.

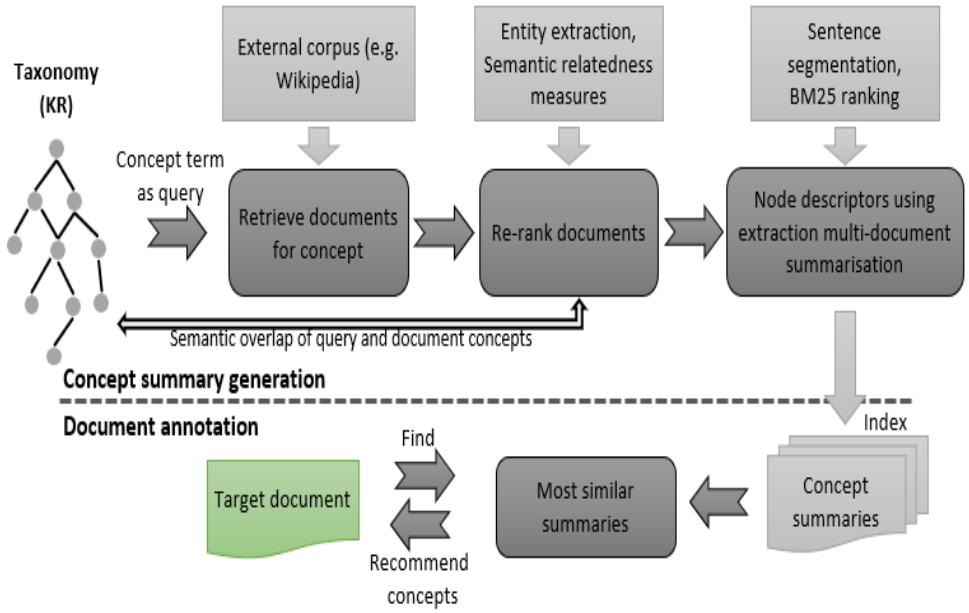


Figure 5.1: Overview of concept summary generation and its use for annotating documents.

5.2.1 Sourcing concept summaries

When generating concept summaries, the objective is to augment the concepts of a KR with useful textual descriptors using an external corpus. First, we identify a set of documents that are potentially relevant to each concept by issuing its label as a query to a corpus. We apply query expansion when there are multiple labels for a concept (e.g. alternative labels or entry terms). Query expansion reformulates a query to include all alternative terms for a concept which is expected to enhance search recall from the target corpus (Manning et al., 2008, Chapter 9). The documents retrieved for a query concept form the candidate sources for generating its summary. We use Wikipedia as the external corpus because it is currently the largest publicly available knowledge repository. Most concepts are expected to have relevant contents within Wikipedia even if they do not have dedicated articles. In other words, although several concepts in a KR may not have corresponding Wikipedia articles, descriptive contents can still be generated using relevant contents from multiple relevant articles. We use the Okapi BM25 to rank Wikipedia articles for each query. BM25 is a state-of-the-art TF-IDF like retrieval function and one of the most successful ranking algorithms for text retrieval (Robertson

et al., 2009). The ranking score of a document d given a query Q containing words $q_1, ..., q_n$ is as shown in equation 5.1.

$$\text{score}(d, Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, d) \cdot (k_1 + 1)}{f(q_i, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \tag{5.1}$$

$f(q_i, d)$ is the term frequency of q_i in d, $|d|$ is the length of d in words, and $avgdl$ is the average document length of the collection from which documents are drawn. k_1 and b are free parameters.

$$\text{IDF}(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} \tag{5.2}$$

$\text{IDF}(q_i)$ is the inverse document frequency of q_i, N is the total number of documents in the collection, and $n(q_i)$ is the number of documents containing q_i.

Concept terms which form the queries are often very short making it difficult to represent an information need (Kwok and Chan, 1998) effectively. Due to causes of ambiguity such as the presence of polysemous terms (e.g. rock: music or stone?), some of the documents retrieved for a concept may not describe it in the sense specified by the KR. Moreover, query expansion as used in identifying candidate documents is known to harm search precision through query drift (Bhogal et al., 2007). Query drift occurs when the terms added through query expansion changes the focus of a query, resulting in the domination of irrelevant documents in the search result. Accordingly, we introduce a semantic re-ranking step to identify a subset of candidate documents that we are more confident of their relevance to a concept.

5.2.2 Context-based filtering of sources

The semantic filtering step measures the degree to which document concepts cluster about a query concept in a taxonomic knowledge resource. The intuition is that a document's relevance to a concept increases as its concepts cluster closer to the query concept. To identify document concepts, we match concept terms from the thesaurus to a keyword index of documents. Both concept terms and the corpus keyword index are stemmed to maximise match discovery. Considering that there may be polysemous terms in the keyword index and the likely introduction of errors by conflating words through

stemming, we require that a concept be only considered present in a document if the document also contains another concept in its *semantic context*. The semantic context of a concept is the set of all concepts which are directly linked to the concept on the taxonomic structure of a KR (Fernández et al., 2011). For example, the semantic context of "rock" in a geological thesaurus may include "igneous rock" and "sedimentary rock". A document that describes the music genre "rock" is unlikely to contain the semantic context of the geological sense of "rock". The outcome of mapping candidate documents to a thesaurus is a bag-of-concepts representation for each document.

Next, we estimate the semantic closeness of each document's concepts to the query concept by cumulating pairwise semantic relatedness measures between query concept and document concepts. The Wu and Palmer algorithm (Wu and Palmer, 1994) is used to measure relatedness between concept pairs. Wu and Palmer measures relatedness based on taxonomic node proximity and depth of entities and its measures correlate well with human judgements of relevance (Hliaoutakis et al., 2006). As shown in equation 5.3, the algorithm preserves the specificity cost and specialisation cost properties which reflects the nature of most taxonomies (Knappe et al., 2007). Specificity cost property requires that the relatedness between neighbouring concepts increase with increasing taxonomic depth. For example, sibling concepts rel("igneous rock","sedimentary rock") should be greater than sibling concepts rel("rock" and "soil") because the former pair, which are narrower concepts of "rock", have greater taxonomic depth. On the other hand, the specialisation cost property requires that further specialisation should imply reduced relatedness. Let $\prec$ denote the broader-than relationship between concepts of such that "rock"$\prec$"igneous rock"$\prec$"felsic" implies that "rock" is the broadest of the three concepts. This property implies that rel("rock", "igneous rock") > rel("rock", "felsic") as "felsic" is a further specialisation of "rock" than "igneous rock". Use of the Wu and Palmer algorithm requires finding the most specific common subsumer (MSCS) of a pair of concepts being compared. The MSCS is the most distant node from the root node that subsumes the concepts.

$$rel(c_1, c_2) = \frac{2 * n(c_1, c_2)}{n(c_1) + n(c_2) + 2 * n(c_1, c_2)} \tag{5.3}$$

c_1 and c_2 are concepts being compared, $n(c_1)$ is minimum node count from c_1 to the MSCS (most specific common subsumer of c_1 and c_2), $n(c_2)$ is minimum node count from c_2 to the MSCS, and $n(c_1, c_2)$ is minimum node count from the MSCS to the root node.

With pairwise semantic relatedness between query and document concepts known, we determine semantic relevance by the semantic relevance of the document by cumulating the relatedness measures as shown in equation 5.4.

$$docScore(d, c_q) = \sum_{c_q, c_i \in C_d} rel(c_q, c_i) \tag{5.4}$$

c_q is the query concept and C_d is the set of concepts of document d. In other words, to determine semantic relevance of a document, the query concept forms a central node on the taxonomy from which we measure the semantic relevance of document concepts. We regard documents with the highest semantic relevance as the most relevant to the query concept.

5.2.3 Extracting concept summaries

The final step of concept summary generation is a multi-document summarisation of the top semantically ranked documents retrieved for each concept. We adopt an extraction-based summary generation approach. This approach extracts the most relevant portions of documents and is similar to the query-biased *snippets* generation used in search environments. Search snippets are short textual summaries of documents that help users to assess the relevance of documents without viewing their entire contents. Query-biased snippets are influenced by the terms in a query and have been shown to be effective in generating useful search snippets (Leal Bando et al., 2015). Instead of a static snippet for each document, query-biased snippets are dynamically generated to be relevant to a query. Accordingly, we generate summaries from documents by ranking their sentences with respect to the query (concept terms) and then selecting the most relevant sentences. We reuse the BM25 ranking function to score the sentences of a document. For this purpose, we treat sentences as documents and treat each document as a collection in applying equation 5.1. Afterwards, summaries (or snippets) of the top K documents are merged to create a concept's summary. We also treat K as a parameter which we experimentally determine in our evaluation.

5.3 Document Annotation using Concept Summaries

The annotation process identifies a subset of concepts from one or more KRs that explicitly describe the content of a document. The annotation task can be modelled as a multi-label classification problem which is formalised as follows. Let $D = \{d_1, \cdots, d_m\}$ represent the collection being annotated and $C = \{c_1, \cdots, c_n\}$ represent concepts in semantic hierarchical KRs from which annotations are selected. Annotation for document $d_j \in D$ using C is a boolean function of the form $f(d_j, C) \rightarrow \mathbf{y^j}$ where $\mathbf{y^j} = [y_1^j, \cdots, y_n^j] \in \{0,1\}^n$ and $y_i^j = 1$ only if $c_i \in C$ is a concept that annotates d_j. In other words, given a document, a binary decision is made on whether each concept in the KR should become part of its annotation.

The first step is to determine a score or probability for each concept in the KR. Also known as concept activation, concept scoring computes scores for concepts which reflect their suitability for annotating a document. Although the annotation process is a binary decision of whether each concept annotates a document or not, the score of a concept (e.g. retrieval score or probability value) represents a confidence value for use in making this decision. As a result, the intermediate output of the annotation function is usually of the form, $\mathbf{y\prime} = [w_1, \cdots, w_n] \in \mathbb{R}^n$ where w_i represents the concept activation for $c_i \in C$.

After concept scoring, an annotation selection method is used to map concept scores $y\prime \in \mathbb{R}$ to a binary decision on annotation $y \in \{0,1\}$ as follows:

$$y = \begin{cases} 1 & \text{if } y\prime \geq \rho \\ 0 & \text{if } y\prime < \rho \end{cases} \quad \forall \quad c_i \in C$$

where ρ is the threshold above which a concept is selected to annotate a document. In semi-automated annotation environments, concepts above the selection threshold are presented to a user to make final selections. As a result, concept recall, which is the proportion of relevant concepts returned by an approach, is an important measure for evaluating annotation performance.

The rest of this section presents two methods for annotating documents with generated concept summaries. These annotation methods are based on concept retrieval and explicit semantic analysis respectively.

5.3.1 Concept retrieval approach

In the concept retrieval approach, the text to be annotated forms a query to indexed concept summaries. To achieve this, we generate TF-IDF vector representations for both the documents being annotated and the concept summaries. Concepts are retrieved for documents by the similarity of vector representations. Once again, we use the BM25 ranking function shown in equation 5.1 to retrieve concept summaries that are most similar to a document. The retrieval score of each concept summary becomes the score of the corresponding concept. This concept scoring approach results in a ranking of concepts so we can impose a threshold t when selecting annotations for each document. In this work, we fix the number of annotations per document because it allows us to compare the performances of different systems after each system has returned the same number of concepts. Accordingly, thresholds $P = \{\rho_1, \cdots, \rho_m\}$ for documents $D = \{d_1, \cdots, d_m\}$ are determined by a top k approach so that a cap applies to the number of concepts that are selected to annotate each document.

5.3.2 ESA approach

Explicit Semantic Analysis (ESA) uses an encyclopedic repository such as Wikipedia to represent terms according to the documents they appear in (Gabrilovich and Markovitch, 2006). We regard each document in the repository as a concept, and the document's content describes the concept. An encyclopedic repository ensures that each document's content focuses on one topic and is expected to be topically orthogonal to other documents. Therefore, content across documents are dissimilar while maintaining a good cohesion between the terms within a document.

The concept-to-terms view is switched such that a vector of concepts represents each term in the vocabulary of the corpus. First, term weights using the TF-IDF weighting scheme gives an association weight between each term and the concept (document). Afterwards, each term is represented by a concept vector composing of its TF-IDF weights in all documents of the corpus. In our approach, the concept summaries form the encyclopedic repository from which we generate concept vectors for vocabulary terms.

Given a term t and collection D, the vector representation for t is $\vec{v_t} = \{\text{tfidf}(t, d_1), ..., \text{tfidf}(t, d_N)\}$ representing the TF-IDF weights of t in every $d \in D$ and each d represents the concept summary of a $c \in \theta$.

$$\text{tfidf}(t, d) = f_{t,d} \cdot \log \frac{|D|}{n_t} \tag{5.5}$$

$f_{t,d}$ is the frequency of t in d, and n_t is the number of documents in D which contain t. This gives a vector representation of length N ($N = |D|$) for each term in the vocabulary of D.

With concept vectors generated, we can represent any piece of text by taking the centroid of the concept vectors of its terms. Annotation using ESA begins by representing the terms of a document by their concept vectors. The concepts are then merged and ranked according to their weights. The best-ranked concepts are selected to annotate the document as described in the concept retrieval approach.

5.4 Semantic precision and recall

Most research works evaluate document annotation systems using standard precision and recall measures for classic IR evaluation as in equations 5.6 and 5.7. The basis for the evaluation measures is the binary decisions on whether each concept returned by an annotation system is correct (present in the gold standard) or incorrect (absent from the gold standard). This binary choice does not account for the quality of returned concepts which are absent from the gold standard. Given the subjective nature of deciding the concept for annotation, another concept that is sufficiently similar to that in the gold standard should not be completely incorrect.

Assuming U represents the annotation concepts for a document in the gold standard, and V represents the concepts that were selected to annotate the document by an approach. Annotation performance using the standard evaluation measures are as follows.

$$P = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{|U_d \cap V_d|}{|V_d|} \tag{5.6}$$

$$R = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{|U_d \cap V_d|}{|U_d|} \tag{5.7}$$

We introduce an evaluation approach which assigns partial scores for concepts depending on how close they are to the gold standard on the KR's taxonomic structure. To illustrate

the motivation for a semantic evaluation, assume we are annotating a document about the fruit "apple" represented as c("apple"). An annotation approach that returns c("fruit") has surely performed better than an approach that returns c("computer") as the document's annotation. With standard evaluation measures, both approaches are deemed to have failed to identify the correct concept and therefore, performed the same. Therefore, instead of counting returned concepts as either 0 or 1 (incorrect or correct respectively) in determining precision and recall, concepts are awarded real-value scores in the $[0, 1]$ range as determined by a semantic relatedness algorithm. A high value indicates that a returned concept is very close to the correct concept, while a low value indicates a wide miss.

Given the gold standard $U = \{u_1 \cdots u_a\}$ and the concepts returned by an annotation system $V = \{v_1 \cdots v_a\}$, we find the optimal pairing between the elements of U and V based on their taxonomic proximity of the ontology. We use the semantic relatedness measure in equation 5.3 to determine relatedness values $rel(u, v)$ (denoted as uv for brevity) for each pair $u \in U$ and $v \in V$ in the cartesian product of U and V as follows:

$$U \times V = \{\, rel(u, v) \mid u \in U \;\text{ and }\; v \in V \,\} = \begin{bmatrix} u_1v_1 & \dots & u_1v_b \\ \vdots & \vdots & \vdots \\ u_av_1 & \dots & u_av_b \end{bmatrix} \tag{5.8}$$

Determining the best pairing in the cartesian product of U and V is a combinatorial optimisation problem, and we use the Kuhn–Munkres algorithm (Hungarian algorithm) to solve this problem. The reason for using the Hungarian algorithm is to ensure that each returned concept is not counted multiple times towards the performance measure of the final evaluation result. The Hungarian algorithm solves an assignment problem by assigning each job to a worker in a manner that minimises the overall cost. The algorithm requires having an equal number of jobs and workers, and the assignment problem is efficiently solved through row and column reductions until it finds the optimal solution. Accordingly, we have to account for two differences for our use case before applying this algorithm. Firstly, we are solving a maximisation problem, and as a result, the additive inverse of matrix elements are used to make it a minimisation problem. Secondly, $|U|$ is not always equal to $|V|$ to form a square matrix. As a result, we adjust the $U \times V$ semantic relatedness matrix to $n \times n$ dimensions with $n = max(|U|, |V|)$. We set the matrix elements without semantic relatedness values to 0 before applying the Hungarian algorithm. After the optimal solution is found, the additive inverse of the minimal cost is gives the

best relatedness pairing which we denote as: $\arg\max \sum_{u \in U, v \in V} rel(u,v) \mid rel(u,v) \leq 1$.

In order to obtain the semantic precision and recall, equations 5.6 and 5.7 are modified as 5.9 and 5.10 respectively. F_{sem} is the harmonic mean of P_{sem} and R_{sem}.

$$P_{sem} = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{\arg\max \sum_{u \in U_d, v \in V_d} rel(u,v)}{|V_d|} \tag{5.9}$$

$$R_{sem} = \frac{1}{|D|} \cdot \sum_{d \in D} \frac{\arg\max \sum_{u \in U_d, v \in V_d} rel(u,v)}{|U_d|} \tag{5.10}$$

5.5 Evaluation

We evaluate the discussed approaches for using corpus-based concept summaries for annotation (CCS) by comparing their performances to alternative annotation approaches on two evaluation datasets.

5.5.1 Datasets and evaluation

We use the Computing and Geology datasets described in section 3.2.2 of Chapter 3. Evaluation uses standard precision (equation 3.4), recall (equation 3.5) and F1 (equation 3.3) measures. Also, we measure the mean average precision (MAP) as shown in equation 3.6. To illustrate the importance of this measure, let us assume two annotation systems have the same precision and recall values after returning a fixed number of concept. The system with a higher MAP value indicates that it ranked the correct concepts higher than the incorrect concepts. The ability to rank the correct concepts higher is preferable because it will make it easier for human annotators to identify the correct concepts when selecting from fewer high-quality concepts. We set $n = 10$ for all approaches for determining MAP and measure the precision, recall, and F1 after returning 5, 7 and 10 concepts for each document.

5.5.2 Experiment setup and alternative approaches for comparison

We use a 5 times 5-fold cross-validation for comparing different annotation approaches. Each time, a random function splits the dataset into a 60% training and 40% testing data. The training data is for use by the supervised approaches and for empirically determining parameters. The supervised approaches use the training data as a pre-annotated corpus for use in annotating the test documents. Concept summaries were generated using 5 best-ranked sentences from 5 most relevant Wikipedia articles as determined by the training data. We obtained comparable results in the region of 2 best sentences from top 2 documents to about 5 best sentences from top 7 documents. Significant reduction in performances was noted when concept summaries grew too large. The performance decrease was determined to be due to added content becoming less useful for distinguishing concepts.

We report results for all approaches based on the ability to reproduce the manual annotations assigned to the testing data. The concepts that were used to annotate documents in the test data form the gold standard. We removed the test data annotations before input to the annotation systems. The external corpus for generating concept summaries is a Wikipedia dump of November 30 2016. Indexing and ranking functions were implemented on Elasticsearch using its Java API[1].

5.5.3 Alternative approaches for comparison

We introduce the following alternative approaches for annotating documents to determine how our approach compares to others. A discussion of annotation approaches and how the techniques employed compare to others was presented in section 2.3. We also provide the parameters used for the annotation approaches in our evaluation.

FREQ: Statistical approach that selects concepts for annotation based on the frequency of concept mentions in a document. Most frequently occurring concepts are selected to annotate each document. FREQ is baseline approach in (Große-Bölting et al., 2015) outperforming several alternative approaches.

HITS: Also uses the co-occurrence graph from DEG but computes scores for each node using the Hyperlink-Induced Topic Search algorithm (Kleinberg, 1999). The selection of nodes with the highest values completes the annotation process. HITS produced the

[1] http://www.elastic.co/guide/en/elasticsearch/client/java-api/5.2

overall best result in comparing different annotation approaches in (Große-Bölting et al., 2015).

DEG: Graph-based approach that first constructs a co-occurrence graph of concepts mentioned in a document. Nodes with the highest degrees (number of edges) are selected to annotate each document. In (Große-Bölting et al., 2015), DEG was very competitive and performed only slightly below the HITS approach.

EAGL: Unsupervised approach that generates concept summaries from textual features of concepts (concept labels, definitions, etc.) (Ruch, 2006). Concept summaries are indexed and BM25 ranking ($k_1 = 1.2$, $b = 0.75$) is used to identify summaries whose concepts should annotate a document. EAGL is the best thesaurus-only annotation system in (Trieschnigg et al., 2009).

BM25: Supervised concept-oriented approach which generates concept summaries using the content of all documents that were annotated with a concept (Trieschnigg et al., 2009). BM25 ranking function ($k_1 = 1.2$, $b = 0.75$) is used to select concepts whose summaries that are most similar to a document.

CLM: Similar to BM25 but uses a language model to discover similar documents (Trieschnigg et al., 2009). We use the Jelinek-Mercer language model which outperformed the Dirichlet model in our experiments. The Jelinek-Mercer uses the Jelinek-Mercer smoothing method (Zhai and Lafferty, 2001). We determined best λ as 0.7.

KNN: Supervised retrieval-based approach which reuses the concepts from k nearest annotated documents to annotate a new document. This method retrieves concepts in the annotation of the nearest documents. Then, the sum of relevance scores of all documents for which a concept formed an annotation determines the concept's score. Although some previous works have used language models to identify nearest documents, BM25 ranking gave the best results in our experiments which we report. The KNN approach is the best-performing annotation approach in the comparative analysis of different approaches in (Trieschnigg et al., 2009).

CCS: CCS is the concept retrieval approach based on concept summaries while CCS_{ESA} is the ESA approach. CCS_{Lite} is a variant CCS which does not re-rank documents that were retrieved for a concept prior to generating its summary. CCS_{Lite} highlights the impact of re-ranking candidate sources of concept summaries in CCS.

5.5.4 Results

Tables 5.1 and 5.2 show the performances of different annotation approaches for the Geology and Computing datasets respectively using standard evaluation measures. The best result for each evaluation metric is in bold.

Table 5.1: Geology: Mean average precision (MAP), macro precision (P), recall (R) and F-measure (F).

	Top 5			Top 7			Top 10			MAP
	P	R	F	P	R	F	P	R	F	
FREQ	0.028	0.085	0.042	0.022	0.089	0.035	0.015	0.089	0.026	0.051
DEG	0.025	0.072	0.037	0.022	0.089	0.035	0.017	0.102	0.029	0.049
HITS	0.024	0.069	0.036	0.021	0.085	0.034	0.015	0.092	0.026	0.048
EAGL	0.111	0.367	0.170	0.090	0.387	0.146	0.075	0.454	0.129	0.230
BM25	0.157	0.500	0.239	0.123	0.532	0.200	**0.100**	**0.598**	**0.171**	0.407
CLM	0.132	0.434	0.202	0.115	0.521	0.188	0.089	0.557	0.153	0.304
KNN	**0.180**	**0.548**	**0.271**	**0.134**	**0.562**	**0.261**	0.099	0.589	0.170	**0.430**
CCS_{Lite}	0.089	0.301	0.137	0.083	0.378	0.136	0.071	0.460	0.123	0.231
CCS_{ESA}	0.115	0.375	0.176	0.097	0.429	0.158	0.080	0.496	0.138	0.271
CCS	0.109	0.364	0.168	0.090	0.407	0.147	0.081	0.523	0.140	0.251

Table 5.2: Computing: Mean average precision (MAP), macro precision (P), recall (R) and F-measure (F).

	Top 5			Top 7			Top 10			MAP
	P	R	F	P	R	F	P	R	F	
FREQ	**0.146**	0.280	**0.192**	**0.119**	0.328	0.175	**0.094**	0.371	0.150	0.161
DEG	0.110	0.211	0.145	0.099	0.264	0.144	0.085	0.332	0.135	0.195
HITS	0.108	0.200	0.140	0.093	0.244	0.135	0.076	0.302	0.121	0.185
EAGL	0.130	0.287	0.179	0.111	0.345	0.168	0.091	0.386	0.147	0.220
BM25	0.097	0.226	0.136	0.093	0.296	0.142	0.076	0.352	0.125	0.201
CLM	0.085	0.202	0.120	0.071	0.236	0.109	0.056	0.267	0.093	0.138
KNN	0.113	0.268	0.159	0.088	0.280	0.134	0.074	0.346	0.122	0.216
CCS_{Lite}	0.123	0.307	0.176	0.104	0.356	0.161	0.091	**0.436**	**0.151**	0.217
CCS_{ESA}	0.123	0.303	0.175	0.099	0.328	0.152	0.081	0.374	0.133	0.209
CCS	0.126	**0.309**	0.179	0.114	**0.393**	**0.177**	0.090	0.427	0.149	**0.224**

We obtained results for our different annotation approaches that are comparable with results in previous works that evaluated their performances (Große-Bölting et al., 2015; Trieschnigg et al., 2009). Supervised approaches (KNN, BM25, and CLM) performed best in the Geology dataset but performed poorly in the Computing dataset. The Geology dataset has a relatively high level of concept reuse for annotation which favours

the supervised methods. CCS and its variants outperformed the other unsupervised approaches on both datasets in most performance metrics. The improvements of CCS over CCS_{Lite} highlights the utility of the concept summary generation strategy. CCS and CCS_{ESA} performed similarly in most cases.

The performances of statistical (FREQ) and graph-based (DEG and HITS) approaches were weak on the Geology dataset, but they showed better performances on the Computing dataset. It appears that the performance difference is due to the nature of the datasets as we will discuss in section 5.6.3. The overall performances for EAGL, CCS, and CCS_{Lite} did not vary considerably between the datasets. Recall that EAGL uses the textual features of a concept as the concept summary. Therefore, the differences between CCS and EAGL are due to the externally generated texts used to augment the textual features of concepts.

Tables 5.3 and 5.4 show the results of applying semantic precision and recall on the same evaluation datasets as Tables 5.1 and 5.2 respectively. Overall performances are higher in the semantic evaluation since it awards partial scores for incorrect annotations depending on how related they are to the correct annotations. The results follow similar trends as previously seen for precision and recall values. Further analysis of the results showed that the variants of CCS returned good quality concepts especially at Top 5 and Top 7 measures. For instance, in the Geology dataset, KNN went from being 50% better than CCS in the standard recall measure to only 20% better in semantic recall at Top 5. Similarly, at Top 10, KNN went from being 38% better than CCS to being only about 14% better. The ability to rank correct concepts higher is expected to make it easier for human annotations since they can assess fewer relevant concepts.

5.6 Discussion

The discussion in this section is based on results of the standard evaluation measures in Tables 5.1 and 5.2.

5.6.1 Annotation performance

As the evaluation results show, automated systems are unable to achieve very high performances when attempting to replicate manual document annotations. The performance

Table 5.3: Geology: Semantic precision (P_{sem}), recall (R_{sem}) and F-measure (F_{sem}).

	Top 5			Top 7			Top 10		
	P_{sem}	R_{sem}	F_{sem}	P_{sem}	R_{sem}	F_{sem}	P_{sem}	R_{sem}	F_{sem}
FREQ	0.051	0.148	0.076	0.040	0.157	0.064	0.028	0.158	0.048
DEG	0.048	0.136	0.072	0.040	0.155	0.064	0.029	0.167	0.050
HITS	0.049	0.135	0.072	0.039	0.152	0.062	0.028	0.159	0.048
EAGL	0.167	0.537	0.255	0.131	0.565	0.212	0.105	0.639	0.180
BM25	0.201	0.634	0.305	0.152	0.661	0.247	0.118	0.723	0.203
CLM	0.186	0.622	0.287	0.150	0.681	0.246	0.116	0.726	0.199
KNN	**0.213**	**0.661**	**0.323**	**0.162**	**0.695**	**0.263**	**0.120**	**0.730**	**0.206**
CCS_{Lite}	0.165	0.537	0.252	0.130	0.575	0.212	0.101	0.627	0.174
CCS_{ESA}	0.167	0.539	0.255	0.134	0.583	0.218	0.108	0.662	0.186
CCS	0.167	0.550	0.256	0.132	0.611	0.217	0.104	0.672	0.181

Table 5.4: Computing: Semantic precision (P_{sem}), recall (R_{sem}) and F-measure (F_{sem}).

	Top 5			Top 7			Top 10		
	P_{sem}	R_{sem}	F_{sem}	P_{sem}	R_{sem}	F_{sem}	P_{sem}	R_{sem}	F_{sem}
FREQ	0.169	0.379	0.234	0.157	0.496	0.238	**0.137**	**0.627**	**0.224**
DEG	0.160	0.367	0.223	0.149	0.479	0.228	0.124	0.571	0.203
HITS	0.167	0.379	0.232	0.148	0.467	0.224	0.128	0.571	0.209
EAGL	0.191	0.455	0.269	0.159	0.521	0.244	0.125	0.589	0.206
BM25	0.162	0.375	0.226	0.153	0.497	0.234	0.123	0.568	0.202
CLM	0.165	0.401	0.234	0.139	0.464	0.214	0.112	0.527	0.185
KNN	0.164	0.393	0.231	0.138	0.452	0.211	0.114	0.527	0.188
CCS_{Lite}	0.183	0.433	0.257	0.158	0.510	0.241	0.128	0.593	0.211
CCS_{ESA}	**0.201**	**0.475**	**0.282**	0.160	0.520	0.244	0.122	0.569	0.201
CCS	0.194	0.471	0.274	**0.161**	**0.538**	**0.248**	0.125	0.590	0.206

limitation is not unexpected since studies have shown that annotation by domain experts can be quite subjective with significant differences in concept selections (Dramé et al., 2016; Trieschnigg et al., 2009). Furthermore, there are upper-performance limits for our evaluation datasets considering that only a few concepts annotate each document. Although precision values are in the 0 to 1 range, the upper limit for precision at top 10 of the Geology dataset (Table 5.1) is 0.18. The limit is because an average of 1.8 concepts annotates each document in the gold standard. Similarly, the upper limit precision for the Computing dataset (Table 5.2) is 0.21. As an illustration, consider a document with two concepts in its annotation as specified in the gold standard. An annotation system

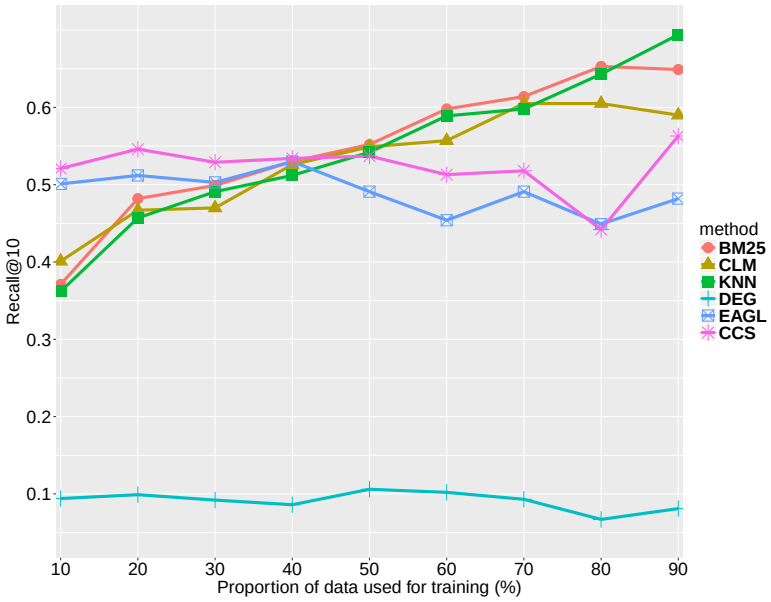
that returns the two correct concepts in its Top 10 selection will only have a precision of 0.2 and F-measure of 0.3. However, recall is the maximum 1.0 indicating that the system discovered that all the relevant concepts. In a semi-automated annotation, a user will have to choose the correct concepts from this subset instead of assessing the entire KR. As a result, recall values are important in the context of this work as it reflects the extent to which an annotation system includes the correct concepts in the returned subset of a KR.

5.6.2 Training-testing dataset splits

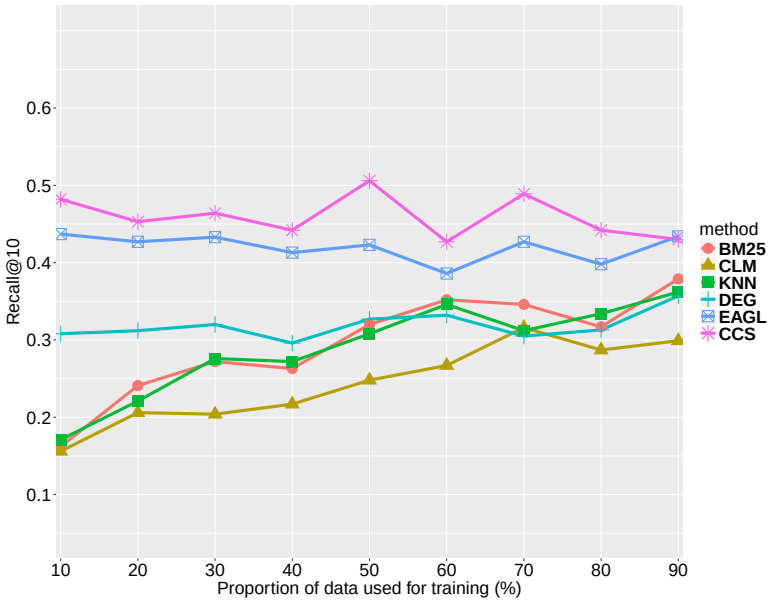
The performances of supervised approaches depend on the amount of data that is available for training. An increase in training data should result in improved annotation performances. The results in Tables 5.1 and 5.2 represent a specific split with the training data (60%) simulating the pre-annotated documents which are used to annotate the held out documents. In Figure 5.2, we show the recall performances of different annotation approaches as the proportion of training data is varied. As expected, the performances of supervised approaches were sensitive to the proportion of data used for training. However, the performances of unsupervised approaches remained relatively steady. In the Geology dataset, CCS was only outperformed by the supervised approaches when we used over 50% of the dataset as the pre-annotated collection. In the Computing dataset, supervised approaches were outperformed by CCS even when 90% of the dataset formed the pre-annotated collection. The reliance of supervised approaches on the quality of a pre-annotated corpus highlights the importance of unsupervised approaches especially in the initial stages of annotation. Although the supervised approaches may perform better when there is sufficient training data, unsupervised approaches remain relevant for generating the initial annotated corpus. Moreover, supervised and unsupervised approaches are usually combined to form hybrid document annotation systems.

5.6.3 Effect of the nature of dataset

As already pointed out, there are some marked differences in performances of some annotation approaches between the datasets. We attribute these differences to the nature of the datasets discussed in section 5.5.1 and summarised by Table 3.2. The performances of statistical and graph-based approaches were better in the Computing dataset than in the Geology dataset, while the supervised approaches were weaker in the Computing dataset.



(a) Geology dataset



(b) Computing dataset

Figure 5.2: Recall with varying proportions of dataset used for training

Overlapping content (i.e. nested document sections) and higher reuse of concepts for annotation appear to have helped the supervised approaches in the Geology dataset. Each concept was reused 2.5 times in the Geology dataset and only 2 times in the Computing

dataset. Greater reuse of concepts increases the likelihood of having them represented in both training and testing data which is expected to benefit supervised approaches. We attribute the difference in performances of graph-based approaches between the datasets to the approach used in generating the gold standards and size of documents. Recall that we generated the annotations for the Computing dataset from keyword tags. Readers assigned most of the keyword tags with the instruction to only select terms appearing in the documents[2]. When assigning keyword tags from document content, it is not unusual to select frequently appearing terms. DEG and HITS appear to have ranked frequently appearing concepts in documents higher since the same concepts form their cooccurrence graphs. A similar pattern in the performances of FREQ supports this observation. On the other hand, each document in the Geology dataset is about 8 times smaller and the text does not often explicitly mention the concept terms used for annotation. The annotations in this dataset were assigned by domain experts who selected concepts from controlled vocabularies based on their understanding of a document's content.

5.6.4 Influence of the semantic re-rank of documents

The improved performance of CCS over CCS_{Lite} highlights the benefit of semantically re-ranking documents before generating concept summaries. As discussed in section 5.2.2, the re-ranking step aims to ensure that we generate sufficiently descriptive texts for concepts. Semantic re-rank is especially useful when concept terms can have multiple meanings. In Table 5.5 are examples of top-ranked sentences that were included in generated concepts summaries by both approaches (CCS and CCS_{Lite}).

In the first two entries of Table 5.5 ("Dyke" and "Ironstone"), clearly, CCS_{Lite} did not generate contents which describe the concepts with respect to a geological domain. In contrast, CCS did generate contents about the concept terms in the required senses. The use of semantic relatedness measures in the re-ranking process allows other entities in a KR to contribute towards determining the best sources for generating concept summaries. Using this semantic disambiguation step, we minimise instances where completely irrelevant textual contents are generated for concepts. The third example, "Appleby Group" is less ambiguous and both approaches generated similar contents for this concept term. Similarly, relevant summaries were generated for "Namurian Age" even though their contents were different.

[2] http://docs.google.com/Doc?id=ddshp584_46gqkkjng4

Table 5.5: Extracts of sample concept summaries

Concept term	Source vocabulary	CCS	CCS_{Lite}
Dyke	GEOTHES	Intrusive ultramafic rocks. The kind of rocks go from conglomerate to shale, volcanic, intrusive and plutonic igneous rocks of many compositions, and metamorphic rocks as well, thus including most major types.	"Bill" Dykes William E. Dykes, Missouri Dykes is an unincorporated community in southwest Texas County, in the U.S. state of Missouri.
Ironstone	GEOTHES	These beds are overlain by the lower Jurassic Lias Group with the Broadford Beds at the base, passing up into the Pabay Shale Formation, the Scalpay Sandstone Formation the Portree Shale Formation and the Raasay Ironstone Formation. Ironstone bands occur in the lower part of the sequence.	Ironstone china Ironstone china, ironstone ware or most commonly just ironstone, is a type of vitreous pottery first made in the United Kingdom in the early 19th century. Ironstone Bank IronStone Bank, was a United States bank that was merged back into First Citizens bank in 2011.
Appleby Group	GEOLEX	The Appleby Group unconformably overlies a variety of older rock strata. It is a basal breccia of cemented limestone and sandstone fragments, dating from the Permian period, forming part of the Appleby Group.	The Appleby Group unconformably overlies a variety of older rock strata. It is a basal breccia of cemented limestone and sandstone fragments, dating from the Permian period, forming part of the Appleby Group.
Namurian Age	GEOCHRON	The Namurian age lasted from 326 to 313 million years ago. The youngest fossils are conodonts which indicate Viséan to Namurian age.	The Namurian age lasted from 326 to 313 million years ago. Lower Coal Measures Formation towards the top of the underlying Stainmore Formation (or Hensingham Formation), which is of Namurian age.

5.7 Chapter Summary

In this chapter, we introduced a corpus-based approach for generating concept summaries (descriptive textual contents) for the entities (or concepts) of a knowledge resource (KR). Our goal was to overcome the limitations of unsupervised document annotation approaches which suffer from sparsity of descriptive textual features for use in annotation. We used knowledge of the taxonomic structure of KRs to identify the best sources for generating concept summaries. With Wikipedia as the external source of concept summaries, semantic relatedness measures were used to filter Wikipedia articles based on their semantic contexts. The motivation for this approach is that when a concept co-occurs with other closely-related semantic entities, there is increased confidence that the article is a reliable source for generating concept summaries. Generated concept summaries were subsequently used to annotate documents using a retrieval-based approach and an ESA approach

Evaluation using two datasets of contrasting features showed that we achieved our objectives and that the semantic ranking process contributed to substantial improvements in some annotation tasks. Analysis of the performance of different annotation approaches in our evaluation highlighted the utility of our approach especially in the initial stages of annotation when there is insufficient training data to support supervised approaches. Supervised document annotation approaches rely on the existence of ample pre-annotated

corpus to annotate new documents. Also, we introduced semantic precision and recall performance measures to estimate the overall quality of the concepts returned by annotation systems. In counting the correctness of the concepts returned by an annotation system, semantic evaluation awards partial to full scores depending on their semantic relatedness to the gold standard concepts. We evaluated annotation systems using the new semantic performance measures and discussed the results.

Chapter 6

Ontology-based Model for Enhancing Search

In information retrieval (IR), the inability to adequately meet information needs by matching keywords alone is well discussed in the literature. The presence of polysemous words (words with multiple senses) results in retrieving irrelevant documents while synonymy (different words but similar meanings) leads to the omission of relevant documents. By representing either or both queries and documents as implicit or explicit semantic entities (concepts), semantic IR approaches aim to retrieve relevant documents even when there is little or no lexical overlap between a query and target documents (Egozi et al., 2011). Ontology-based IR techniques are explicit concept-based approaches and are particularly suitable for search in specialised domains where the concepts described by the ontology are not in everyday language use (Choi et al., 2016). It is indeed easier to build and maintain comprehensive ontologies for specific domains than for all knowledge areas. Semantic Web technologies have increased the availability of both hand-crafted and automatically generated ontological knowledge resources and their use in IR.

Domain ontologies enable retrieval systems to better "understand" domain specific terms in queries and documents. It regards all the terms used to describe a concept (i.e. its alternative realisations) as synonyms for retrieval. Other entities which are linked to a concept specify its semantic context which is useful for disambiguation. Taxonomic relations specify hyponyms (more specific terms than those given) and hypernyms (more generic terms than those given) of concepts. These subsumption relations can be leveraged to identify documents which describe concepts that are not directly mentioned by a

query but are closely related to it. Such consideration is especially useful in exploratory search where the search intent is to retrieve multiple relevant documents. As an example, consider a user who wants to find documents about "apple" (the fruit). Documents about "Rosaceae" (family of apple-like plants) or "Dabinett" (an apple cultivar) may be deemed close enough to the search intent. Also, some documents may refer to "apple" by alternative names such as "Malus pumila" (the scientific name for apple). Even more, the knowledge that "apple" is related to "Rosaceae" and "Dabinett" can help to filter out non-relevant documents such as those about "Apple" (the company) given that they are unlikely to contain those related terms.

6.1 Problem Definition

Since domain ontologies only model sub-areas of knowledge, they are commonly used as part of hybrid retrieval systems to add semantic dimensions to search. With ontology-based IR viewed as adding a semantic layer to existing search methods, experimental results show that while the semantic layer can improve the overall performance of an underlying search system, semantic considerations do not benefit some queries (Castells et al., 2007; Fernández et al., 2011). It is challenging to maintain a high search precision since introducing new terms (adding generalisation, specialisation, or synonyms of query terms) to match more documents in the search system often results in query drift. Adding generic terms to precise queries is unlikely to result in significant performance improvements. Also, semantic retrieval may not be helpful when there is insufficient ontological coverage for concepts that are relevant to a query. It remains a challenge to reliably tell when the use a semantic retrieval approach will be beneficial for a retrieval task. Most of the work on the use of ontological knowledge resources for document retrieval focuses on query representations alone. The use of ontological knowledge to influence the scoring function for document ranking has received little attention. For example, in query expansion, queries are augmented with relevant terms from an ontology, but the document matching process often uses traditional keyword-based search methods (Lu et al., 2009).

In this chapter, we introduce a SelecTive Ontology-based Retrieval Model (*STORM*) which is a hybrid model that uses the conceptual overlap between query and document to add a semantic layer on existing search methods. More specifically, we use ontology knowledge to influence document matching in the concept space. In doing so, we use the knowledge of equivalent terms, hypernyms and hyponyms of a query in a semantic

ranking algorithm. We do not envisage that such semantic considerations will always enhance search performance. Therefore, we hypothesise that the attributes of a query and the ontological context of the concepts it expresses can help in determining when semantic ranking will benefit a retrieval instance. Accordingly, *STORM* includes a predictive model which relies on features that are extracted from a query and relevant domain ontology to determine when semantic retrieval will be beneficial for a query. We restrict query features to those that apply to enterprise search systems where the document base is domain specific (e.g. biomedicine, geoscience) and the existence of links between the documents such as hyperlinks found on Web pages, cannot be guaranteed. Advances in enterprise search systems have not kept up those of Internet search engines. Often, the retrieval techniques in Internet search engines are not directly applicable to enterprise systems, or they lead to poor results (Li et al., 2014). We assume the existence of ontologies that are of a similar domain as the target collection and that the ontologies specify taxonomic relations between concepts (e.g. is-a, broader/narrower-than) forming hierarchical concept graphs (HCGs). For ease of explanation, we use the term "ontology" to represent knowledge resources with taxonomic relations such as controlled vocabularies, thesauri and semantic nets, and "concept" to represent semantic entities that are specified by the knowledge resources.

Our main contributions are (i) a hybrid retrieval model that semantically ranks documents based on the semantic overlap between query and document concepts and (ii) a predictive model that signals when to use semantic ranking based on the predicted benefit to a query. The hybrid retrieval model incorporates a keyword-based component which our semantic contribution aims to enhance. Accordingly, the main research questions are:

1. Can the ranking of documents based on the degree of semantic overlap between query concepts and document concepts enhance retrieval performance?

2. Can we learn to predict when semantic ranking will benefit or not benefit retrieval for a query?

6.2 Ontology-based Retrieval Model

SelecTive Ontology-based Retrieval Model (*STORM*) is a hybrid retrieval model that selectively uses semantic ranking for document retrieval. Central to *STORM* is a domain

ontology which we use for the conceptual representation of both queries and documents. The conceptual representations are the basis for semantic ranking. The decision on whether to use semantic ranking for a query depends on its predicted benefit which is in turn, informed by pre-retrieval features which we generate for each query. Figure 6.1 is a high level view of *STORM* showing its key components. We summarise the retrieval approach in the following steps:

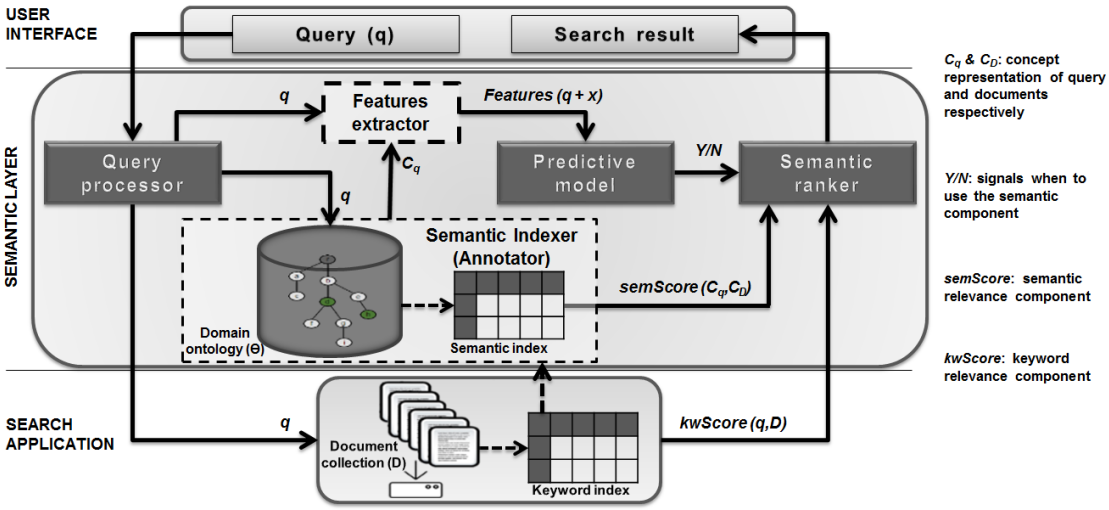


Figure 6.1: Overview of the selective ontology-based retrieval model (*STORM*) as a semantic layer on a search application to improve relevance ranking.

1. Query processor maps query to ontology concepts to produce query concepts. This is a query annotation step to identify the concepts mentioned by a query. Section 6.2.1 describes the query mapping process.

2. A predictive model signals whether to use semantic ranking based on features generated for the query. We discuss this in section 6.3.

3. When using semantic ranking, the semantic ranker measures the semantic overlap between query concepts and document concepts using a semantic relatedness algorithm.

4. Semantic and keyword-based document relevance scores are aggregated to determine search result ranking. Section 6.2.3 describes steps 3 and 4.

Note that the steps outlined above only apply to queries that contain one or more ontology concepts as determined in the query processing stage. *STORM* skips the semantic layer if a query does not map to any concepts. It also skips the semantic layer whenever the

predictive model signals that semantic ranking will not enhance the search effectiveness of a query. Without using the semantic layer of *STORM*, retrieval defaults to the base retrieval system – a keyword-based model in this case. The keyword-based ranking uses the keyword index while semantic ranking uses the semantic index. The semantic index is generated by document annotation as described in section 6.2.2

6.2.1 Query Processing

The query processing step analyses free text natural language queries to identify ontology concepts that they contain. We adopt an approach for mapping free text queries to knowledge resources in (Meij et al., 2011) as follows.

1. Generate n-grams of contiguous words in the query and sort them in descending order of length.
2. Starting from the longest n-gram, match each n-gram to textual labels of ontology concepts.
3. If an n-gram matches a concept label, remove the n-gram from the query string and re-generate n-grams from the remaining query terms.
4. Repeat the process in steps 1 to 3 above until we can no longer map query n-grams to concept labels.

To maximise concept discovery, we stem words to overcome mismatch due to word inflexion (e.g. disease vs diseases). However, stemming can also conflate terms thereby increasing the likelihood of matching unintended concept labels. As a result, we expect a user to validate query concepts and possibly modify them before use for document retrieval.

The query mapping process is described using a portion of MeSH ontology shown in Figure 6.2 and query "treatment for mad cow disease" (see Table 6.1). All n-grams of contiguous words are generated from the query and sorted in decreasing order of length while preserving the order of appearance (i.e. left to right) when n-grams are the same length. Matching starts from the longest n-gram because of the general assumption that a longer term is more specific and therefore, a preferred match when compared with possible matches of its substrings (Castells et al., 2007). The matched concept, c_7 has two text labels "Bovine spongiform encephalopathy" and "Mad cow disease" as specified

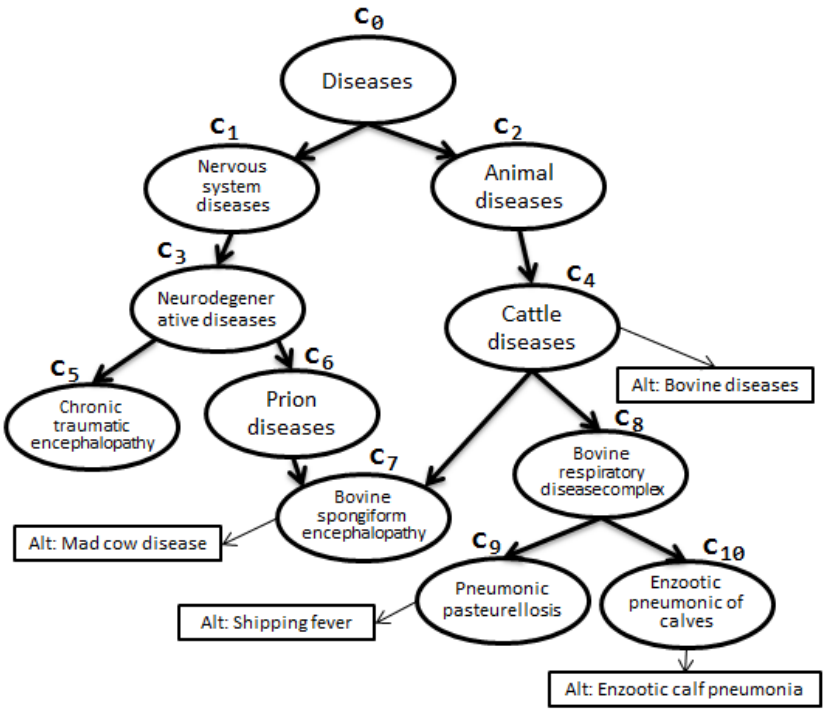


Figure 6.2: Extract of MeSH with ellipses showing preferred text labels for concepts and arrows indicating direction of increasing specialisation in the concept hierarchy. Texts in rectangles represent alternative entry terms for concepts (synonyms).

Table 6.1: n-grams generated for query, q ("treatment for mad cow disease") sorted and mapped to ontology concepts in Figure 6.2. The mapping approach identifies concepts that are present in a query by matching generated query n-grams to the text labels of concepts.

Sorted $ngram(q)$	Matched concepts
treatment for mad cow disease	–
treatment for mad cow	–
for mad cow disease	–
treatment for mad	–
for mad cow	–
mad cow disease	c_7 {labels: Bovine spongiform encephalopathy, Mad cow disease}
treatment for	–
treatment	–
for	–

by MeSH. The use of either text labels in q will map to the same concept, potentially overcoming the synonymy problem. Note that by matching from the longest n-gram, we rightly match c_7 to q instead of the less specific c_0 ("Diseases"). After c_7 was matched, its term "mad cow disease" was removed from the original query with n-grams re-generated from the remaining query terms. No other n-grams could match concept terms after c_7.

6.2.2 Semantic Indexing

STORM is composed of a keyword index and a semantic index. The keyword index is based on a bag-of-words representation of documents with TF-IDF weighting and uses the vector space model (VSM) for keyword-based retrieval. The semantic index, on the other hand, is based on a bag-of-concepts representation of documents by mapping documents to the concepts they contain. We determine the weight of concepts in the semantic index at retrieval time. The semantic relatedness measure between concepts in the semantic index and the query concepts determines concept weights. Section 2.6 discusses the concept of semantic relatedness and the next section (section 6.2.3) will discuss the weighting of concepts in more detail.

In order to generate the semantic index of *STORM*, we annotate documents by adopting an approach used in Fernández et al. (2011). Concept terms are looked up in a keyword index of the collection to identify documents that contain each concept. Terms from both the collection's keyword index and ontology are stemmed to maximise concept discovery. Also, we only annotate a document with a concept only if the document contains both the concept and an entity in its semantic context. The semantic context of a concept is the set of all semantic entities which have a direct link to the concept in the ontology. The requirement for the presence of a semantic context is due to the possible presence of polysemous terms in the keyword index and the likely introduction of errors in annotation through stemming. Illustrating with Figure 6.2, the semantic context of c_7 is the set of directly linked concepts $\{c_4, c_6\}$. Therefore, if "mad cow disease" or "Bovine spongiform encephalopathy" appears in a document, c_7 is only added to its bag-of-concepts representation if the document also contains either "Prion diseases" or "Cattle diseases". The reinforcement approach is expected to enhance annotation precision with some loss in recall. Table 6.2 shows an example of bag-of-concepts representation of documents in collection D.

At this stage, the potential for the use of the ontology and semantic index to semantically

Table 6.2: Semantic index indicates ontology concepts that are present in a document.

Document	Bag-of-concepts
d_1	c_2, c_4, c_5
...	...
$d_{\|D\|}$	c_3, c_5

enhance document retrieval over a collection is as follows:

1. The requirement for the presence of a semantic context for concept identification in documents is expected to minimise the polysemy problem (false-positive problem) through its sense disambiguation ability.

2. The use of alternative terms to describe a concept as specified by the ontology will alleviate the synonymy problem (false-negative problem) by abstracting from alternative realisations to concepts.

3. Taxonomic relations indicating hypernyms and hyponyms will help to identify semantically related contents even when there is minimal direct concept overlap between query and documents.

Although Figure 6.1 shows the keyword index is used to generate the semantic index, it is only depicted this way for ease of presentation. In reality, a separate stemmed index that treats concept terms like individual tokens is created to enable the matching of phrasal terms that appear in an ontology.

In generating the semantic indexing, there is also the option of using the semantic annotation approach with corpus-based concept summaries introduced in Chapter 5. The semantic annotation approach limits the conceptual representation of a document to a few concepts that are deemed to be highly relevant. However, annotation requires having a relevant document collection with which to generate the concept summaries. Such collections are not easy to come by when dealing with specific concepts in specialised domains as we observed from our evaluation dataset. Also, deciding the number of concepts that sufficiently represent the content of a document is not trivial. We will expect the number to vary between documents.

6.2.3 Semantic Document Ranking

The degree of semantic overlap between document concepts and query concepts determine the semantic document ranking in *STORM*. The motivation for this approach is that a document's relevance to a query increases as its concepts cluster closer to query concepts on the concept lattice of an ontology. This way, a document whose concepts are sufficiently related to the concepts of a query is considered semantically relevant even if the document does not contain any query concepts.

Semantic Relatedness between Concepts

The pairwise semantic relatedness measures between the query concepts and document concepts provide an estimate of the semantic closeness of a query to a document. There are a variety of algorithms for measuring the semantic relatedness between the concepts of an ontology (Blanchard et al., 2005). We agree with the requirement that an appropriate semantic relatedness algorithm should preserve both specificity cost property and specialisation cost property. The specificity cost property requires that the relatedness between neighbouring concepts increase with increasing taxonomic depth (Knappe et al., 2007). Using the taxonomic relations between concepts in an HCG, the proximity and taxonomic depth of concepts can be readily determined. Following the specificity cost property, we expect specific terms "Pneumonic pasteurellosis" and "Enzootic pneumonic of calves" to be closer related than "Nervous system diseases" and "Animal diseases" since the former pair have greater taxonomic depth. That is, $rel(c_1, c_2) < rel(c_9, c_{10})$ in Figure 6.2. The depth of a concept is its distance from the top or root concept following specialisation relations. The specificity cost property also ensures that specialisation is more desirable than generalisation. Assuming search intent is c_4, more specific concepts c_7 and c_8 should be closer related to c_4 than a more generic concept like c_2. The specialisation cost property requires that when comparing a concept to other concepts, further specialisation implies reduced semantic relatedness. For example, if a query is c_4 and there are documents about c_8 and c_9 in the collection. Based on these concepts alone, this property implies that the document about c_8 is more relevant than the document about c_9 since relative to c_4, c_9 is a further specialisation than c_8.

We use Lin's algorithm (Lin, 1998) to measure semantic relatedness between concepts as it satisfies required cost properties. Lin's algorithm, as shown in equation 6.1, uses

a combination of relative positions of concepts on an ontology's structure and the information content of concepts to estimate semantic relatedness between concepts which correlates well with human judgments of relatedness (Hliaoutakis et al., 2006).

$$rel(c_i, c_j) = \frac{2 * logP(m(c_i, c_j))}{logP(c_i) + logP(c_j)} \tag{6.1}$$

c_i and c_j is the pair of concepts being compared, $m(c_i, c_j)$ is the most specific common subsumer of c_i and c_j, $P(c) = freq(c)/N$ is the information content of concept c, $freq(c)$ is the corpus frequency of c added to the corpus frequency of all other concepts it subsumes, and N is the total count of concepts in the corpus.

Equation 6.2 cumulates pairwise relatedness measures between query concepts and document concepts to determine the semantic relevance score of a document.

$$semScore(d, q) = \frac{\sum_{c_i \in C_d, c_j \in C_q} rel(c_i, c_j)}{\sum_{c \in \Theta, c_j \in C_q} rel(c, c_j)} \tag{6.2}$$

Θ is the domain ontology, $C_d \in \Theta$ is the set of concepts used to annotate document d, and $C_q \in \Theta$ is the set of concepts mapped to q. The denominator is the normalisation factor which is the maximum relevance score obtainable assuming a document contains all the concepts that are related to the query concepts on the ontology.

A demonstration of the process for computing the semantic relevance score of a document is presented in Figure 6.3. Each concept that is triggered by a query forms a focal point from which the semantic relatedness of document concepts are determined. In other words, the extent to which the concepts expressed in a document cluster close to the query concepts determine semantic document relevance. Semantic relatedness measures between concepts form a matrix whose density depends on the degree of concept interconnectedness on the ontology. In $STORM$, the pairwise semantic relatedness between concepts are pre-computed and persisted in memory as coordinate lists $(c_i, c_j, rel(c_i, c_j))$. We look up the coordinate lists for relatedness measures when a query becomes available at retrieval time.

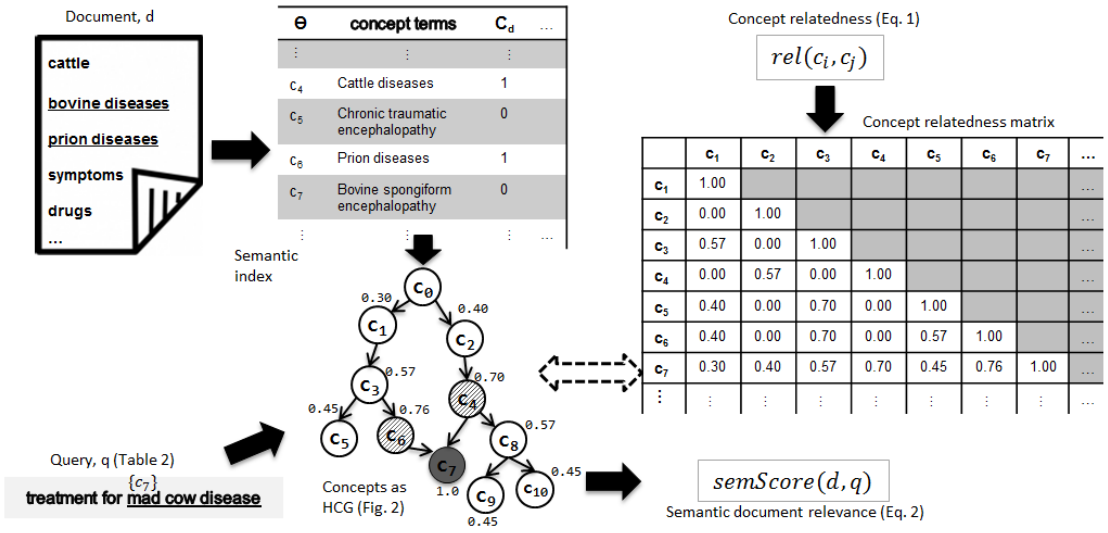


Figure 6.3: Demonstration of the semantic ranking process of *STORM* which measures the semantic overlap between the conceptual representation of query and documents.

Aggregation of Relevance Scores

The semantic relevance score of each document is aggregated with the corresponding keyword-based relevance score to determine retrieval order. Different ontology-based IR approaches use a similar aggregation of relevance scores (Castells et al., 2007; Fernández et al., 2011; Paralic and Kostial, 2003). We aggregate scores for two main reasons.

1. To account for incomplete ontological knowledge. It is often the case that not all query terms map to ontology concepts. The keyword-based component compensates for this since it deals with queries at the term level.

2. To minimise the impact of imprecise annotation of queries and documents. Any automated approaches for semantic annotation or entity linking are imprecise.

The vector space model (VSM) is used to determine the keyword-based relevance of documents. VSM measures the relevance of documents at term level based on the TF-IDF weights of query terms contained in the documents. TF-IDF weights are computed for terms as shown in equation 6.3 to determine the components of the vector representation of documents. The cosine similarity of document vectors and query vectors, as expressed in equation 6.4, determines the keyword-based relevance score of a document. We envisage the ability to substitute the VSM approach with alternative keyword-based retrieval approaches.

$$\text{tf-idf}_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{1 + \text{df}_t} \tag{6.3}$$

$\text{tf-idf}_{t,d}$ is the TD-IDF weight of term $t \in d$, $\text{tf}_{t,d}$ is the frequency of $t \in d$, df_t is the number of documents in the collection containing t, N is the total number of documents in the collection.

$$kwScore(d, q) = \frac{\vec{q} \cdot \vec{d}}{||\vec{q}||||\vec{d}||} \tag{6.4}$$

$kwScore(d, q)$ is the keyword-based relevance score of d to q, $\vec{d}$ and $\vec{q}$ are the vectors of d and q respectively.

Next, the relevance scores aggregation step linearly combines $semScore(d, q)$ and $kwScore(d, q)$ as shown in equation 6.5. We adopt a linear combination rather than multiplicative aggregation which can make a component to have an overwhelming effect on a document's rank. The linear combination approach is an effective method of aggregating the results of multiple searches (Fernández et al., 2011; Fox and Shaw, 1994). We normalise $semScore(d, q)$ to be in the same range as $kwScore(d, q)$ prior to aggregation.

$$docScore(d, q) = \alpha * semScore(d, q) + (1 - \alpha) * kwScore(d, q) \tag{6.5}$$

where $\alpha \in [0, 1]$ is the weight attached to each sub-expression and is experimentally determined. In the subsequent step, $\alpha = 0$ whenever the predictive model signals that semantic ranking will not benefit a query. We discuss the predictive model next.

6.3 Predictive Model for Semantic Ranking

Previous works have shown that semantic document ranking, as part of hybrid retrieval systems, does not improve the retrieval performance of every query. While semantic ranking improves the retrieval performances for some queries, the performance for other queries remain unchanged or are even worse off. We hypothesise that the intrinsic features of a query and its conceptual context can predict when ontology-based semantic ranking will not improve a retrieval task. Semantic ranking as described in section 6.2.3 may not be beneficial for retrieval in the following situations:

1. Retrieval for unambiguous queries. We expect the keyword-based method alone to be effective when retrieving documents for very specific queries.

2. Insufficient ontological coverage of concepts that are relevant for retrieval. With sparse coverage of concepts, remotely related concepts are assigned higher than expected semantic relatedness measures due to their taxonomic proximity.

The ability to use semantic ranking only when it will enhance the retrieval performance of a query will further improve the overall performance of *STORM*. We model the decision on whether to use or not use semantic ranking as a classification problem. The queries whose retrieval performance are enhanced by semantic ranking form the positive class while remaining queries form the negative class. *STORM* uses the semantic layer when retrieving documents for queries in the positive class and skips the semantic layer when retrieving documents for the negative class. Accordingly, a set of features are generated to characterise each query. Given a query's features, the decision on whether to use semantic ranking is made by a predictive model which is trained on problem instances by supervised learning. The rest of this section describes the features that are used to characterise queries and the predictive model of *STORM*.

6.3.1 Query Features

The query features are crucial for building the predictive model which determines when to use or not semantic ranking. The decision on query features build on Meij et al. (2011) which describes features for queries for discovering mappings to a knowledge resource. We group query features into n-gram features and concept features as summarised in Table 6.3. N-gram features are generated from query terms while concept features are generated from the ontological context of query concepts. We can determine all the query features before any documents are retrieved (i.e. pre-retrieval query features), and thus minimises the performance overhead of using the predictive model in a retrieval system.

N-gram Features

Length of query (*NGram*): This is the number of words in a query with stop words removed and whitespace characters indicating word boundaries (He and Ounis, 2004). The intuition for including this feature is that queries with fewer words are generally

Table 6.3: Features of query, q with query concepts, C_q.

	Features		Description
n-gram	$NGram$	$len(q)$	Length of query.
	$CCover$	$len(q')/len(q)$	Proportion of query terms that are mapped to ontology concepts.
	$CCount$	$\|C_q\|$	Number of query concepts.
concept	$MinDepth$	$\min_{c \in C_q} depth(c)$	Minimum depth of query concepts.
	$MaxDepth$	$\max_{c \in C_q} depth(c)$	Maximum depth of query concepts.
	$AvgDepth$	$\sum_{c \in C_q} depth(c)/\|C_q\|$	Average depth of query concepts.
	$MinPFreq$	$\min_{c \in C_q} \|sup(c)\|$	Minimum frequency of parent concepts.
	$MaxPFreq$	$\max_{c \in C_q} \|sup(c)\|$	Maximum frequency of parent concepts.
	$AvgPFreq$	$\sum_{c \in C_q} \|sup(c)\|/\|C_q\|$	Average frequency of parent concepts.
	$MinCFreq$	$\min_{c \in C_q} \|sub(c)\|$	Minimum frequency of child concepts.
	$MaxCFreq$	$\max_{c \in C_q} \|sub(c)\|$	Maximum frequency of child concepts.
	$AvgCFreq$	$\sum_{c \in C_q} \|sub(c)\|/\|C_q\|$	Average frequency of child concepts.
	$Neighbours$	$\sum_{c \in C_q} (\|sup(c)\| + \|sub(c)\|)$	Frequency of directly linked concepts.

considered to be less specific and hence more ambiguous than longer queries (Kwok and Chan, 1998).

Proportion of query mapped to concepts ($CCover$): This feature represents the proportion of a query that maps to the concepts of the ontology. Specifically, this is the ratio of the length of query terms that map to ontology concepts to the length of the entire query. In Table 6.3, q' represents the query terms that are mapped to ontology such that $q' \subseteq q$. As a degree of overlap, this is a real value bounded in $(0, 1]$ with 1 indicating that the entire query maps to concepts and values closer to 0 indicating minimal ontological coverage.

Number of query concepts ($CCount$): This is the number of concepts that map to a query. This feature can impact on semantic ranking since each concept that maps to a query form a point from which we measure the semantic relatedness between query concepts and document concepts. Recall that $STORM$ skips the use of semantic ranking if a query does not map to any concepts. That is, when $CCount = 0$ or $CCover = 0$.

Concept Features

Depth of query concepts: The taxonomic depth of concepts can indicate the general or specific nature of a query considering that concepts become increasingly specific with increasing depth on an HCG. Accordingly, we generate three features which are functions of the depth of query concepts. These are features representing the minimum depth of query concepts ($MinDepth$), the maximum depth of query concepts ($MaxDepth$), and the average depth of query concepts ($AvgDepth$).

Neighbourhood of query concepts: The density of concept clusters usually vary between portions of an ontology. One reason for such imbalance is the incomplete knowledge representation of an ontology, and this is expected to impact the use of an ontology for semantic ranking. Since the most semantically related concepts to another concept are those in its immediate ontological neighbourhood, we include the frequency of concepts that are directly linked to query concepts ($Neighbours$) in the feature set. We further separate directly linked concepts as parent concepts and child concepts using subsumption relations. Thus, additional features are generated from the neighbourhood of query concepts to represent the minimum count of parent concepts ($MinPFreq$), the maximum count of parent concepts ($MaxPFreq$) and the average count of parent concepts ($AvgPFreq$). We make this distinction because it can reflect additional information about the ambiguity of query concepts. For example, the sense of a concept with multiple parent concepts is more diluted than the sense of a concept with one parent. Similarly for child concepts, we generate features to represent the minimum count of child concepts ($MinCFreq$), the maximum count of child concepts ($MaxCFreq$) and the average count of child concepts ($AvgCFreq$).

6.3.2 Predictive Model

Figure 6.4 is an overview of the building and application of a predictive model for deciding when to use or not use semantic ranking for document retrieval. Search query terms and the concepts that map to a query form the input to a features extractor. The features extractor generates the features to characterise a query as described in section 6.3.1.

In the training phase, the machine learning algorithm learns to classify query instances based on their features and associated class labels ("Y" or "N"). Semantic ranking is beneficial to retrieval (class label = "Y") whenever adding the semantic component in

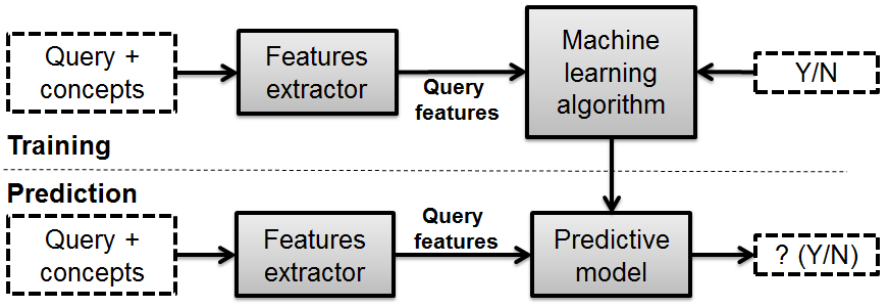


Figure 6.4: Predictive model uses features extracted for each query to decide when to use semantic ranking.

equation 6.5 improves its average precision. Otherwise, the use of semantic ranking does not benefit retrieval (class label = "N"). Subsequently, the learned model is used to determine the utility of semantic ranking for unseen queries.

6.4 Evaluation

We experimentally compare *STORM* with its components, to analyse the influence of selective semantic search enhancement, and with alternative retrieval approaches. This section discusses the experiment setup and outcome.

6.4.1 Experiment setup

We compare the performance of *STORM* with alternative retrieval approaches using two TREC datasets (the 2006 and 2007 genomic tracks) and MeSH as described in section 3.2.3 of Chapter 3. The evaluation collection consists of 162,259 documents which we semantically indexed with about 70 million MeSH concepts (16,933 unique). Fifty-three of the 62 queries (85.5%) mapped to 120 concepts (70 unique) making an average of 1.92 concepts per query. We use this subset of queries with at least, one query concept for the evaluation since *STORM* does not attempt to semantically enhance queries that do not map to any ontology concepts.

We use five times cross-validation, and each time, we order the queries by topic numbers and sample every ith entry ($i \in [1, 5]$) to generate a training sample of about 80% for parameter tuning and a test sample of about 20%. Using the training samples, α in

equation 6.5 was determined as 0.25 for all folds by grid search. The training samples were also used to select the best parameters for alternative retrieval approaches in our evaluation. In measuring MAP for performance measure, we set $n = 100$ for all systems. n is the maximum number of top-ranked search results that are evaluated. We test for significance in performance differences using paired t-test on AP values at 95% confidence interval.

6.4.2 Alternative retrieval approaches for comparison

Alternative retrieval approaches compared in this evaluation are as follows:

- VSM_{TFIDF}: Baseline approach which is keyword-based retrieval. We implemented a VSM approach with TF-IDF weights and cosine similarity for relevance ranking using the Apache Lucene library. While generating the keyword index, we remove stopwords consisting of 33 most commonly used English language words as specified by Lucene and stored term position offsets to enable phrase search. VSM_{TFIDF} also forms the keyword retrieval component of $STORM$ and corresponds to performance results when $\alpha = 0$.

- $BM25$: This is the Okapi BM25 implementation and represents state-of-the-art keyword retrieval approach. We use $k1 = 1.2$ and $b = 0.75$ which have been empirically determined to be suitable parameters for most collections (Manning et al., 2008). $BM25$ was also implemented using Apache Lucene library.

- QE: This approach performs a query expansion using synonyms extracted from query concepts. The identification of query concepts uses the same approach as $STORM$. After query expansion, we perform a keyword-based search using the new query as in VSM_{TFIDF}.

- VSM_{CFIDF}: This approach applies VSM to the concept space by adapting TF-IDF for concept representations. We also implemented SEIR ($VSM_{CFIDF} + VSM_{TFIDF}$) which is a hybrid approach that aggregates VSM_{CFIDF} and VSM_{TFIDF} to represent the approach described in (Fernández et al., 2011). We determined aggregation weight using the test sample as 0.1 for VSM_{CFIDF} and 0.9 for VSM_{TFIDF}.

- SEM_{REL}: This approach represents semantic document retrieval using semantic relatedness measures between the query concepts and document concepts.

SEM_{REL} is the semantic component of $STORM$ and represents the performance of $STORM$ with $\alpha = 1$ for all queries. We also obtain retrieval performance of ORM ($SEM_{REL} + VSM_{TFIDF}$) which represents a variant of $STORM$ without the predictive model. In other words, this variant applies semantic ranking when retrieving documents for all queries.

6.4.3 Results and discussion

MAP in Table 6.4 shows that ORM enhanced the performance of VSM_{TFIDF} but the overall improvement was only 2.7%. The moderate improvement is because by indiscriminately applying semantic ranking, ORM did not improve the performance of VSM_{TFIDF} in 36% of queries (see Table 6.5). The semantic component worsened the retrieval performances for some Topics such as 208 and 229. Figure 6.5 shows how the AP of ORM differs from VSM_{TFIDF}. Positive differences show performance improvements while negative differences show performance decreases due to applying semantic ranking. Apart from Topic 184 with 800% positive difference and excluded for better visualisation, performance differences range from +75% to -30%. We treat Topic 184 as an outlier and exclude it from subsequent analysis. $STORM$ attempts to avoid instances of negative performances by predicting non-beneficial and possibly harmful use of semantic ranking.

Table 6.4: Mean average precision (MAP) of retrieval approaches with ↑ and ↓ indicating significant difference in AP from VSM_{TFIDF} ($p < 0.05$).

	MAP
VSM_{TFIDF}	0.2114
$BM25$	0.2161
QE	0.1545↓
VSM_{CFIDF}	0.1198↓
SEM_{REL}	0.1542↓
$SEIR$ ($VSM_{CFIDF} + VSM_{TFIDF}$)	0.2113
ORM ($SEM_{REL} + VSM_{TFIDF}$)	0.2171

VSM_{CFIDF} and SEM_{REL} relied on conceptual representations only and performed poorly. Conceptual representations alone leads to poor results as not all aspects of a query can be represented in a conceptual language (Trieschnigg et al., 2009). $SEIR$ and ORM demonstrate why semantic approaches are usually hybrid systems as they improved on the performances of VSM_{CFIDF} and SEM_{REL} respectively. Besides, we cannot return documents for queries that are not mapped to any concepts when relying on conceptual representations alone. We point out that since VSM_{CFIDF} depends on the

Table 6.5: Average Precision (AP) on individual queries. The best value for each topic is displayed in bold font face.

Topic	VSM_{TFIDF}	$BM25$	QE	$SEIR$	ORM
160	0.1373	0.1582	**0.3639**	0.1380	0.1618
161	0.3037	**0.3463**	0.0115	0.2771	0.2740
165	0.2892	**0.4907**	0.1162	0.289	0.3351
166	0.2807	0.1552	0.0831	0.2824	**0.3094**
167	0.2929	**0.4352**	0.0027	0.2933	0.2803
170	**0.4000**	0.1410	0.3333	**0.4000**	**0.4000**
171	0.0029	**0.0294**	0.0033	0.0029	0.0029
172	0.0061	**0.0077**	0.0040	0.0066	0.0072
175	0.2472	**0.5341**	0.2800	0.2604	0.2694
178	0.0559	0.0465	0.0078	0.0559	**0.0602**
179	0.0694	**0.0907**	0.0789	0.0710	0.0831
181	**0.4045**	0.3230	0.3150	0.4043	0.4035
183	0.1696	0.1899	0.1622	0.1710	**0.1979**
184	0.0185	0.0192	0.0263	0.0185	**0.1667**
185	0.3789	**0.4907**	0.3058	0.3780	0.4313
186	0.3312	0.3404	0.1366	0.3313	**0.3419**
200	0.1892	0.1429	0.1674	0.1880	**0.2058**
201	0.0929	**0.1933**	0.0542	0.0952	0.1164
202	**0.0900**	0.0243	0.0677	0.0871	0.0638
203	0.4059	0.3824	0.3943	0.4046	**0.4088**
205	0.1142	0.1246	0.0539	0.1113	**0.1298**
206	0.4400	0.3771	0.4346	**0.4406**	0.4287
208	**0.2627**	0.2467	0.2354	0.2623	0.2157
211	0.3578	0.2290	0.3582	**0.3621**	0.3378
213	0.3069	0.2986	0.1238	0.3069	**0.3123**
214	0.2973	**0.4068**	0.1703	0.2962	0.3115
215	0.2751	0.2547	0.0650	0.2731	**0.3012**
216	0.0345	**0.0421**	0.0140	0.0341	0.0398
218	0.2528	0.1695	0.0052	0.2534	**0.2624**
219	0.0567	0.0610	0.0540	0.0565	**0.0730**
220	0.7774	0.7510	0.7285	**0.8016**	0.6527
223	0.0766	**0.3809**	0.1482	0.0744	0.0914
226	0.2505	0.0903	**0.4271**	0.2414	0.2104
227	0.1058	0.0286	0.0489	0.1058	**0.1062**
228	0.0045	0.0020	**0.0060**	0.0042	0.0038
229	**0.4804**	0.4694	0.2033	0.4788	0.4550
231	0.0403	0.0249	0.0082	0.0402	**0.0492**
232	0.0668	0.0233	**0.1248**	0.0667	0.0641
233	0.0400	0.0655	0.0007	0.0400	**0.0701**
234	0.0480	0.0557	**0.0585**	0.0482	0.0496

numeric statistics of concepts in a corpus, it is expected to be relatively sensitive to the method of document annotation. The presence of co-references and symbols/acronyms makes it difficult to obtain accurate counts of concept mentions in natural language text. In contrast, SEM_{REL} relies on a boolean concept identification (present or absent) and

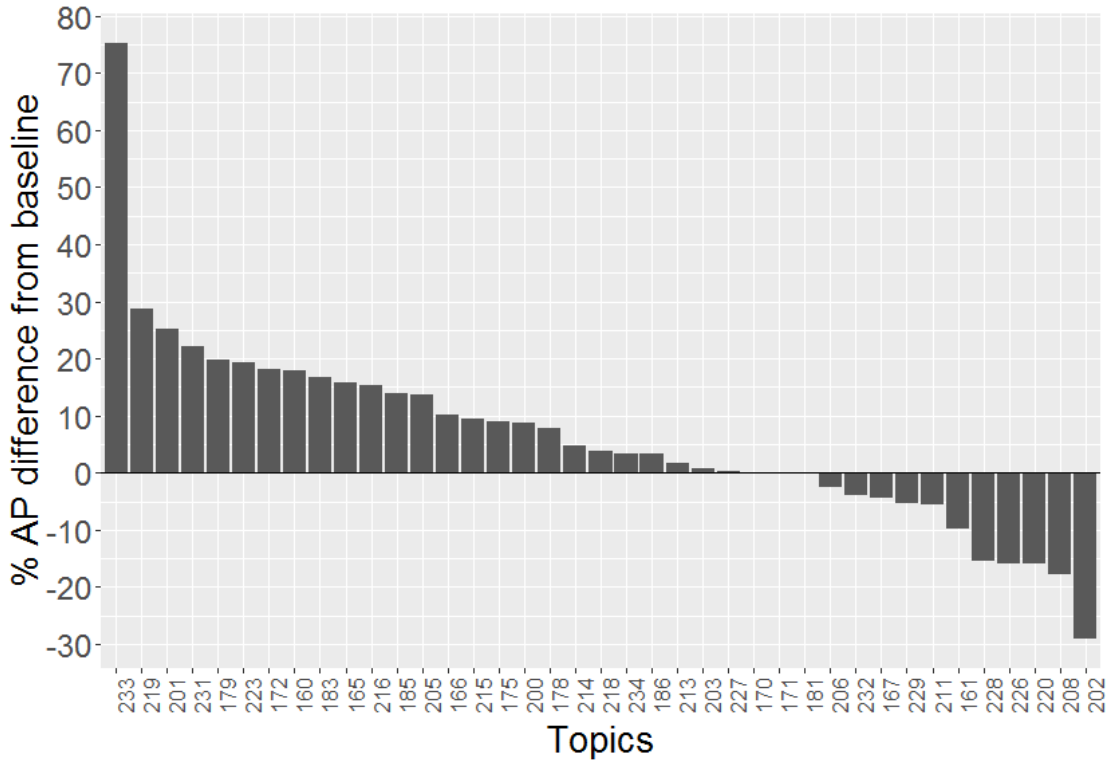


Figure 6.5: Percentage differences in Average Precision (AP) of retrieval instances for ORM compared to VSM_{TFIDF}.

is thereby, less sensitive to the annotation method. Other results in Table 6.4 show that the performance of $BM25$ is comparable to ORM and is better than VSM_{TFIDF} (2.2% better) while QE was weak (27% worse than VSM_{TFIDF}).

Selective Semantic Ranking

We use a predictive model to signal when to use semantic ranking in $STORM$ as described in section 6.3. Each query, with its feature vector (attributes), forms an instance in the machine classification problem. In training the classifier for our evaluation, we use the output of ORM, which applies semantic ranking to all queries, to determine class labels. The beneficial use of semantic ranking corresponds to retrieval instances where the AP of ORM improves on VSM_{TFIDF}. Otherwise, we consider semantic ranking to be non-beneficial for retrieval.

Classifiers and Parameters

We compare prediction accuracy on four machine classifiers: Decision Tree (J48); k-Nearest Neighbours (kNN); Naive Bayes (NB); and Support Vector Machine (SVM). The WEKA Java library[1] was used to build the classifiers. Due to the limited size of problem instances, we use a leave-one-out cross-validation (LOOCV) to estimate the performance of each classifier. In LOOCV, an instance is in turn, left out while the rest of the problem instances forms the training set used to build a classifier. The class of the left-out instance, which simulates an unseen query, is then predicted with the model in the test phase. Accordingly, we use the average of test instances of LOOCV predictions to estimate the retrieval performance of *STORM* for each classifier. The main classifier parameters used are as follows.

- J48: Pruned tree, confidence factor: 0.25, and minimum instances per leaf 3.
- kNN: $k = 5$ and Manhattan Distance as the distance function for Nearest Neighbour search algorithm.
- NB: No supervised discretisation nor kernel estimator.
- SVM: Used the LibSVM library[2] with radial basis function (RBF) as kernel type. SVM type: C-SVC, degree: 3, cost: 1000, gamma: 0.005.

Classifier Results

Classification results are summarised in Table 6.6. F-measure is the harmonic mean of precision and recall. FP Rate shows the false positive rates. Since this is a binary classification problem, random assignment of class labels should have about 50% accuracy. Classification with ZeroR had 64% accuracy while OneR based on the average depth of query concepts (*AvgDepth*) had 68% classification accuracy (OneR minimum bucket size of 5). NB performed best with an overall accuracy of 82% in differentiating between beneficial and non-beneficial use of semantic ranking. The ROC Area and Kappa statistic highlight the slight superior performance of NB when compared to the other classifiers.

We update retrieval performances according to MAP in Table 6.4 to include *STORM* as shown in Table 6.7. Reported result for *STORM* uses the NB classifier. *STORM*

[1] http://www.cs.waikato.ac.nz/ml/weka/
[2] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

Table 6.6: Summary result of classification algorithms on *WUP* using leave-one-out cross-validation.

	Precision	Recall	F-Measure	FP Rate	ROC Area	Kappa
J48	0.807	0.800	0.802	0.210	0.705	0.5763
KNN	0.805	0.800	0.798	0.283	0.736	0.5318
NB	0.824	0.820	0.812	0.271	0.788	0.5841
SVM	0.800	0.800	0.800	0.234	0.755	0.5660

enhanced the performance of VSM_{TFIDF} even further by being able to avoid most of the negative performances of ORM. Although it is a 6% increase in MAP, the improvement of $STORM$ from VSM_{TFIDF} is statistically significant ($p < 0.05$). $STORM_{max}$ is the upper bound performance of $STORM$ assuming 100% classification accuracy on the evaluation dataset.

Table 6.7: Mean average precision (MAP) of retrieval approaches with ↑ and ↓ indicating significant difference in AP from VSM_{TFIDF} ($p < 0.05$).

	MAP
VSM_{TFIDF}	0.2114
$BM25$	0.2161
QE	0.1545↓
VSM_{CFIDF}	0.1198↓
SEM_{REL}	0.1542↓
$SEIR\ (VSM_{CFIDF} + VSM_{TFIDF})$	0.2113
$ORM\ (SEM_{REL} + VSM_{TFIDF})$	0.2171
$STORM$	0.2236↑
$STORM_{max}$	0.2256↑

Varying Training – Testing Proportion

We had used LOOCV to generate the performance result of $STORM$ which is an average performance of multiple classifiers. Here, we build and test one NB classifier for comparison with LOOCV result by splitting our dataset into training and testing samples. We also vary the proportions of training and testing samples to observe the classifier's sensitivity to the size of the dataset. Accordingly, a classifier is built using a proportion of the dataset and tested with the rest. Table 6.8 shows classification accuracy and other performance indicators.

WEKA was used to randomly split the dataset into training and testing samples (randomisation seed = 1). As the results indicate, classification accuracy did not vary significantly until the proportion of the training sample went below 50% of the dataset. The

Table 6.8: Result of NB classifier varying train–test data split. (PRC is precision-recall curve.)

Train-Test	F-Measure	ROC Area	PRC Area	Kappa	Accuracy
80% - 20%	0.800	0.792	0.818	0.583	80.0%
70% - 30%	0.860	0.759	0.777	0.706	86.7%
60% - 40%	0.793	0.823	0.857	0.565	80.0%
50% - 50%	0.798	0.780	0.789	0.576	80.0%
40% - 60%	0.618	0.431	0.549	0.162	63.3%

relative stability in performance suggests that we can use few problem instances to build a good classifier model for $STORM$.

Semantic Ranking and Query Features

Recall that the only difference between ORM and $STORM$ is that ORM does not use a classifier to differentiate queries. Accordingly, we consider the queries at the performance margins of ORM to see if we can ascertain why the indiscriminate use of semantic ranking benefits some queries but not others relative to VSM_{TFIDF}. One of the worst performing queries for semantic ranking was Topic 202 ("What drugs are associated with lysosomal abnormalities in the nervous system?") with 30% decrease in AP. This query mapped to 2 concepts for terms "drugs" and "nervous system" both at depth 1 of the MeSH hierarchy. Considering that concepts nearer the root node tend to have broader senses, it appears like semantic ranking caused a drift from the query's topic. Also, no concepts map to a key entity of the query "lysosomal abnormalities" which might have improved semantic ranking. At the other extreme, the semantic component improved the AP of Topic 233 ("What viral genes affect membrane fusion during HIV infection?") by 75%. This query maps to 3 concepts for terms "viral genes", "membrane fusion" and "HIV infection". These concepts are neither too broad nor too specific as they are all at depths 3 or 4 on MeSH. Also, there is a high overlap between the query terms and MeSH. These queries suggest some form of interplay between the features of a query and the effect of the semantic ranking on retrieval performance. However, an analysis of the entire evaluation dataset shows that we cannot draw simplistic conclusions as to how query features affect semantic ranking.

In Figure 6.6, we show the average performance differences of ORM from VSM_{TFIDF} according to query features: query length ($NGram$), query overlap with ontology ($CCover$), taxonomic depth of query concepts ($AvgDepth$), and number of parent concepts of query concepts ($AvgPFreq$). Dots on the scatter plots represent queries, and the

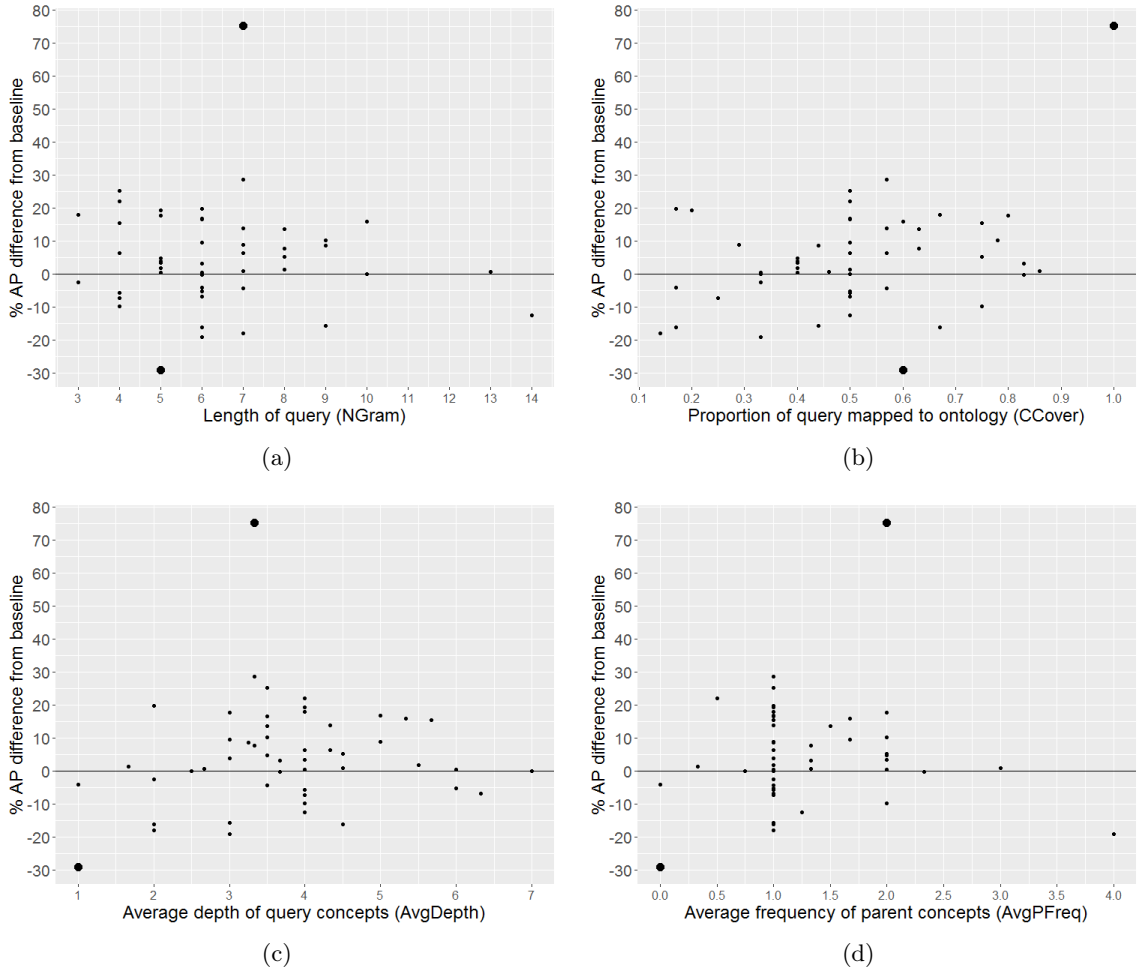


Figure 6.6: Scatter plots showing the performance difference of ORM from VSM_{TFIDF} (baseline) according to different query features. None of the query features seem to correlate with retrieval performance.

horizontal axes are corresponding feature values. Emphasised dots are the two queries at the performance margins which we discussed (topics 202 and 233). The patterns show no clear indication of how query features affect the use of semantic ranking. Most of the best performances correlate with queries with concepts at an average depth between 3 and 6, but that is also where most of the data lie (Fig. 6.6(c)). Performance according to query length (Fig. 6.6(a)) agrees with previous works that have not found a clear correlation with retrieval performance (He and Ounis, 2004). Indeed, the analysis of retrieval performances according to query features shows that an individual feature cannot be used to reliably determine when semantic ranking is beneficial confirming the need

for several features for generating our classifier.

6.5 Chapter Summary

In this chapter, we introduced *STORM*, a semantic retrieval model for selective search enhancement. *STORM* consists of keyword-based and semantic-based search components. The semantic component of *STORM* uses the semantic overlap between concepts expressed in queries and documents for document ranking. When indiscriminately applied to all queries, semantic ranking worsens retrieval performance for some queries. Accordingly, a classifier-based approach was used to selectively apply semantic ranking to queries which it is predicted to benefit. The predictive model of *STORM* was able to predict when it was beneficial to apply semantic ranking with reasonable accuracy based on collection-independent pre-retrieval query features only. Evaluation using TREC datasets showed that *STORM* could enhance search effectiveness of a keyword-based retrieval system and comparison with alternative search approaches showed promising results.

Chapter 7

Conclusion

In this thesis, we addressed the problem of document retrieval in enterprise systems using domain ontologies. We identified three main challenges for the successful use of domain ontologies for document retrieval. The first challenge is how to achieve a broad ontological domain coverage which we addressed through the alignment of multiple overlapping domain ontologies. The second problem is entity linking whereby free-text documents and queries are annotated by the entities (or concepts) specified in ontologies. We presented a novel method for annotating documents or its segments using ontology concepts. The third is the use of conceptual representations of documents and queries to semantically rank documents. We introduced a document retrieval framework to enhance search performance by incorporating a semantic ranking function and a predictive performance model in the explicit semantic space provided by domain ontologies. In the rest of this chapter, we examine the extent to which our research objectives were achieved in addressing these challenges and outline areas for future extensions of this work.

7.1 Contributions

Develop effective methodologies for the alignment of knowledge-light ontologies by incorporating the ability to infer semantic correspondences

In chapter 4, we presented novel methods for aligning ontologies using supervised and unsupervised approaches. The motivation for this objective is that ontology alignment allows for the substitution of single ontologies with multiple aligned ontologies thereby

providing a broader domain coverage for information systems. Ontology alignment systems predominantly use string similarity techniques with the assumption that similar concepts will have similar lexical representations (Cheatham and Hitzler, 2013). String similarity fails to discover alignment when concepts have different lexical representations such as synonyms. As a result, semantic similarity techniques attempt to discover alignment correspondences by meaning, but limits on the vocabulary size of semantic approaches make them unable to discover other correspondences which string-based techniques can discover. The effective combination of string-based similarity and semantic similarity techniques remains a challenge in implementing alignment systems (Otero-Cerdeira et al., 2015; Shvaiko and Euzenat, 2013). Furthermore, the integration of structural information such as details of the semantic context of concepts, in an alignment system can uncover alignment correspondences which direct concept comparisons alone are unable to discover.

Our supervised ontology alignment approach, $Rafcom$ (Random Forest Classifier-based Ontology Matching) integrates string-based similarity, semantic similarity, and structural features to build a random forest classifier model. In $Rafcom$, ontology alignment is viewed as a classification problem such that when presented with the features of a pair of concepts from different ontologies, a classifier model determines if the concepts are aligned concepts or not aligned concepts. The uniqueness of $Rafcom$ includes the introduction of novel similarity features for the machine classifier and the incorporation of word embedding for semantic match discovery in the alignment process. Experimental evaluation using benchmark datasets from the Ontology Alignment Evaluation Initiative (OAEI) showed that $Rafcom$ outperformed state-of-the-art alignment systems while relying on minimal information from the ontologies. Analysis of the performance of $Rafcom$ highlighted how introduced features contributed to the discovery of different types of alignment correspondence.

Considering that training data is not always available for supervised ontology alignment, we introduced two unsupervised ontology alignment approaches WHS (Weighted Hybrid Similarity) and WVS (Weighted Vector Similarity). WHS and WVS use a hybrid of string-based similarity, semantic similarity using word embedding vectors, and term weighting to discover alignment correspondences between concepts. The main distinction between both approaches is in the method of combining their components. WHS uses a known method for text (phrases or sentences) similarity which finds the best similarity coupling between individual terms from the texts being compared. It uses a hybrid of semantic similarity and string similarity to measure the element-level similarity between

concepts. WHS also integrates term weights using the TF-IDF weighting scheme to determine the importance of individual words in phrasal concept terms. WHS's approach can be viewed as an extension of Soft-TFIDF (Cohen et al., 2003) with the hybrid similarity technique as its base similarity method. Another feature to the hybrid similarity technique of WHS is the limit placed on the contribution of string-based similarity. A threshold was specified below which string similarity does not count towards the overall similarity of concepts. In the WVS approach, the magnitude of word embedding vectors of individual words in concept terms are scaled according to their TF-IDF term weights. Scaled vectors of words are then aggregated for phrasal concept terms before use for the discovery of alignment correspondences. Our evaluation showed that manipulating word representations in the vector space led to improved performances in the use of word embedding for alignment. Although WHS and WVS were outperformed by $Rafcom$ in the evaluation, their performances were comparable with the best systems from the OAEI challenge.

Both the supervised and unsupervised alignment approaches presented rely on minimal information from the ontologies making them suitable for aligning knowledge-light ontologies. Most of the commonly used ontologies in knowledge organisation systems are lightweight such as MeSH and AGROVOC[1]. Lightweight ontologies lack some several features that are available in well-formalised ontologies such as having a clear separation of the A-Box and T-Box, the presence of data properties and object properties, and specifying value restrictions.

Propose a framework for the semantic annotation of documents that can deal with sparse descriptive textual features in lightweight ontologies

The use of ontologies for concept-based document retrieval requires linking the contents of documents to the concepts specified in ontologies. The most effective methods for annotating documents with ontology concepts rely on reusing concepts that were assigned to similar annotated documents. However, the problem of generating this initial set of annotated content has to be solved when using such methods. Also, it is challenging to annotate with concepts which do not appear in the initial annotated set when reusing them to annotate new documents. Alternative approaches that do not require a pre-annotated corpus usually rely on textual features in ontologies (e.g. concept terms, synonyms, definitions). These often lead to poor results as most ontologies lack sufficient

[1] http://aims.fao.org/vest-registry/vocabularies/agrovoc

descriptive textual features for concepts that can be used to match them to document contents effectively.

In chapter 5, we proposed an unsupervised method for annotating documents (or segments of a document) based on generating concept descriptors from an external resource. Specifically, we augmented the concepts of an ontology with descriptive textual features (concept summaries) that were sourced from Wikipedia. An essential process in our approach is the novel method of determining the most relevant content in the external resource for generating concept summaries. In determining relevant sources, we used the knowledge of semantic relatedness between the concepts of an ontology to identify documents from which to extract concept summaries. Our approach assumes that the concepts of an ontology form a taxonomy which is used as a semantic filter to disambiguate mentions of concept terms in documents. Subsequently, we explored two methods of using generated concept summaries for the unsupervised annotation of documents. The first annotation method uses a term vector representation of concept summaries and target documents. This method treats the annotation of documents as a concept retrieval task. The second annotation method applies explicit semantic analysis (ESA) to concept summaries to obtain a concept vector representation for each term in the collection of concept summaries. Concepts in the centroid of the vectors for all the terms of a document become the document's annotation. Our evaluation showed that the proposed annotation methods using generated concept summaries outperformed other unsupervised approaches and were comparable with the supervised approaches. The result highlights the utility of proposed approaches in the initial stages of an annotation task when there are no annotated documents for the supervised approaches.

Finally, we discussed the limitations of using standard precision and recall measures for evaluating semantic annotation systems and proposed semantic precision and recall measures. The main difference between the standard and semantic evaluation approaches is the method of determining the correctness of the concepts returned by an annotation system. The standard evaluation using makes a binary decision on correctness while semantic evaluation measure how close a returned concept is to the correct concept. We argued that semantic evaluation provides a better measure for determining the performance of an annotation system.

Develop a novel semantic ranking algorithm that maximises use of domain knowledge in ontologies for document retrieval

In chapter 6, we presented a semantic ranking function that uses ontology concepts for document ranking. Unlike statistical and vector-based approaches, our semantic ranking function is based on the hypothesis that the relevant documents to a query should contain concepts that are highly related to the query concepts. With both document and query mapped to ontological concepts, the ranking function retrieves documents based on how query concept cluster close to document concept on the ontology concept lattice. The Wu and Palmer (1994) algorithm was used to measure semantic relatedness between concepts based on their relative positions on the ontology structure. Experiments showed that this relatedness-based approach outperformed a commonly used approach which adapts the vector space model by replacing term vectors with concept vectors.

Investigate use of supervised machine learning to predict when to semantic ranking will be beneficial for document retrievals

The review of previous works on the use of ontologies to enhance search performance revealed that ontology-based semantic ranking does not enhance the retrieval quality for all queries. While retrieval quality for some queries are improved compared alternative retrieval approaches, the retrieval quality for other queries remain unchanged or become worse. The use of conceptual representations for retrieval is resource-intensive, and it is preferable to avoid its use when not beneficial for document retrieval. We explored the ability to predict when to use or not use semantic ranking to prevent unnecessary or harmful use of semantic ranking. In section 6.3 of chapter 6, we introduced a predictive model for determining queries whose retrieval performance will be improved by semantic ranking. We characterised queries using a combination of previously known query features and novel features generated from the query string and the ontology. Based on the query features, a naive Bayes classifier was able to determine when to use or not use semantic ranking with 82% accuracy.

Propose a semantic document retrieval framework which integrates the semantic ranking and predictive performance models

With a semantic ranking model based on ontology concepts and a predictive model that determines when to use or not use semantic ranking, this objective integrates both models in a document retrieval system. Chapter 6 introduced a document retrieval framework, *STORM* which adds the semantic ranking model as a layer on the keyword-based vector space model. When a query is entered, *STORM* uses the predictive model to determine whether the semantic model will enhance the retrieval performance relative to the base retrieval system. *STORM* only uses the semantic layer when it is predicted to improve retrieval performance and skips it otherwise. Experimental evaluation using TREC genomic datasets and the MeSH ontology showed that both the semantic ranking model and predictive model contributed to significantly improving the performance of *STORM* over the base retrieval system. Also, *STORM* outperformed alternative ontology-based document retrieval systems in our evaluation.

7.2 Future Work

This section highlights the limitations of the work presented in this thesis and outlines some areas for consideration in future extensions. First, in the ontology alignment approaches which we introduced, the machine learning approach ($Rafcom$) uses two main stages by first identifying candidate alignment correspondences and then determines whether each candidate correspondence is an actual correspondence (positive class) or a false correspondence (negative class). Identifying candidate alignment correspondences use four similarity measures, and the choice of the similarity thresholds influence the class balance when generating the machine classifier. When the thresholds are set very high, the positive class tends to outnumber the negative class, and this trend reverses when the similarity thresholds are set very low. Systematic determination of similarity thresholds and investigating how to deal with class imbalances when generating the classifier model will improve the system's robustness. Also, the ability to transfer a trained model to a different domain will be beneficial especially in the initial stages of alignment when there are no reference alignments with which to generate a classifier. In the unsupervised ontology alignment approaches, incorporating structure-level matching techniques and introducing post-alignment refinements using a reasoner are expected to improve overall performance. Structure-level features such as the similarity between the semantic

contexts of entities being compared can reveal additional correspondences which direct comparisons alone cannot discover.

On the semantic annotation of documents, generated concept descriptors were used to annotate documents using a document retrieval approach. Future consideration will investigate alternative ways of using concept summaries for annotating documents. An example is a graph-based approach in Hulpus et al. (2013) which is potentially applicable by using concept descriptors as encyclopedic knowledge resource such as Wikipedia. Also, when annotating segments of documents, we assumed the independence of each segment by treating it as a separate document. In reality, the rest of the document from which a segment is extracted can provide useful contextual information for determining the right concepts with which to annotate the segment. It might be beneficial to treat segments of documents differently from entire documents.

Another consideration for future extensions is to investigate the generalisation or transferability of *STORM* to other domains. Building the predictive model of *STORM* requires training data which may not be readily available in some domains. The ability to transfer a trained model across domains will enhance the utility of *STORM*. Also, current semantic relatedness measures for comparing concepts rely on concept proximity on hierarchical concept graphs (HCGs). Missing intermediate nodes affect the quality of relatedness measures obtained which can in turn impact on ontology-based semantic ranking. Semantic relatedness approaches which do not rely on the structure of HCGs can overcome this problem. Preliminary work using word embedding vectors to measure the relatedness of concepts showed promising results. The major setback is that several domain ontologies do not have concept terms in the vocabulary of the corpus for generating word embedding models. Recent developments in techniques for generating embedding vectors for out-of-vocabulary terms are promising for using word embedding. Also, instead of a binary classification on when to use or not use semantic ranking, a possible extension of *STORM* is to use a regression model that predicts how much of the semantic relevance component to use for each retrieval task. It will require more data points than was available for this research to build such a model.

There is also the challenge of determining how many concepts in the ontologies are sufficient for achieving good performances in semantic document retrieval. How the number of concepts impact on retrieval performance can be estimated empirically by iteratively increasing the number of concepts used for semantic ranking while noting the impact on retrieval performance. In a setting with multiple ontologies, we can increase

the number of concepts by merging new ontologies through alignment. Finally, another enhancement is the use of the relationships between the entities in ontologies to provide more semantic content to users in *STORM*. Entity relationships can be used in bespoke ways to meet information needs better. These include providing additional information as is done by Google using the Knowledge Graph. Entity relationships can help user navigation and lead to the discovery of useful information, especially in exploratory search.

Bibliography

Andrenucci, A. and Sneiders, E. (2005). Automated question answering: Review of the main approaches. In *Third International Conference on Information Technology and Applications (ICITA'05)*, volume 1, pages 514–519. IEEE.

Aronson, A. R. (2001). Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. In *Proceedings of the AMIA Symposium*, page 17. American Medical Informatics Association.

Aronson, A. R., Mork, J. G., Gay, C. W., Humphrey, S. M., and Rogers, W. J. (2004). The NLM indexing initiative's Medical Text Indexer. *Medinfo*, 11(Pt 1):268–72.

Athukorala, K., Głowacka, D., Jacucci, G., Oulasvirta, A., and Vreeken, J. (2016). Is exploratory search different? a comparison of information search behavior for exploratory and lookup tasks. *Journal of the Association for Information Science and Technology*, 67(11):2635–2651.

Berlanga, R., Nebot, V., and Pérez, M. (2014). Tailored semantic annotation for semantic search. *Web Semantics: Science, Services and Agents on the World Wide Web*.

Bernard, L., Einspanier, U., Haubrock, S., Hübner, S., Klien, E., Kuhn, W., Lessing, R., Lutz, M., and Visser, U. (2004). Ontology-based discovery and retrieval of geographic information in spatial data infrastructures. *Geotechnologien Science Report*, 4:15–29.

Bhogal, J., Macfarlane, A., and Smith, P. (2007). A review of ontology based query expansion. *Information Processing & Management*, 43(4):866–886.

Bigot, A., Déjean, S., and Mothe, J. (2015). Learning to choose the best system configuration in information retrieval: The case of repeated queries. *Journal of Universal Computer Science*, 21(13):1726–1745.

Blanchard, E., Harzallah, M., Briand, H., and Kuntz, P. (2005). A typology of ontology-based semantic measures. In *Enterprise Modeling and Ontologies for Interoperability EMOI-INTEROP*.

Blanco, R., Cambazoglu, B. B., Mika, P., and Torzec, N. (2013). Entity recommendations in web search. In *International Semantic Web Conference*, pages 33–48. Springer.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Brill, E., Dumais, S., and Banko, M. (2002). An analysis of the askmsr question-answering system. In *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing-Volume 10*, pages 257–264. Association for Computational Linguistics.

Broder, A. (2002). A taxonomy of web search. In *ACM SIGIR Forum*, volume 36, pages 3–10. ACM.

Budanitsky, A. and Hirst, G. (2006). Evaluating wordnet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

Buscaldi, D., Rosso, P., and Arnal, E. S. (2005). A wordnet-based query expansion method for geographical information retrieval. In *CLEF (Working Notes)*.

Carpineto, C. and Romano, G. (2012). A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1.

Castells, P., Fernandez, M., and Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 19(2):261–272.

Chauhan, R., Goudar, R., Sharma, R., and Chauhan, A. (2013). Domain ontology based semantic search for efficient information retrieval through automatic query expansion. In *Intelligent Systems and Signal Processing (ISSP), 2013 International Conference on*, pages 397–402. IEEE.

Cheatham, M. and Hitzler, P. (2013). String similarity metrics for ontology alignment. In *International Semantic Web Conference*, pages 294–309. Springer.

Choi, J., Park, Y., and Yi, M. (2016). A hybrid method for retrieving medical documents with query expansion. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 411–414. IEEE.

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pages 73–78.

Collins, A. M. and Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, 82(6):407.

Croft, W. B. (2000). Combining approaches to information retrieval. In *Advances in Information Retrieval*, pages 1–36. Springer.

Cronen-Townsend, S., Zhou, Y., and Croft, W. B. (2004). A framework for selective query expansion. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 236–237. ACM.

Cruz, I. F., Antonelli, F. P., and Stroe, C. (2009). AgreementMaker: Efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, 2(2):1586–1589.

Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM.

David, J., Euzenat, J., Scharffe, F., and Trojahn dos Santos, C. (2011). The alignment API 4.0. *Semantic web*, 2(1):3–10.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Dragoni, M., da Costa Pereira, C., and Tettamanzi, A. G. (2012). A conceptual representation of documents and queries for information retrieval systems by using light ontologies. *Expert Systems with Applications*, 39(12):10376–10388.

Dramé, K., Mougin, F., and Diallo, G. (2016). Large scale biomedical texts classification: A kNN and an ESA-based approaches. *Journal of Biomedical Semantics*, 7(1):40.

Egozi, O., Markovitch, S., and Gabrilovich, E. (2011). Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems (TOIS)*, 29(2):8.

Euzenat, J., Shvaiko, P., et al. (2007). *Ontology Matching*, volume 333. Springer.

Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., and Couto, F. M. (2013). The AgreementMakerLight ontology matching system. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 527–541. Springer.

Fernández, M., Cantador, I., López, V., Vallet, D., Castells, P., and Motta, E. (2011). Semantically enhanced information retrieval: An ontology-based approach. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):434–452.

Figueroa, A. and Atkinson, J. (2016). Ensembling classifiers for detecting user intentions behind web queries. *IEEE Internet Computing*, 20(2):8–16.

Fox, E. A. and Shaw, J. A. (1994). Combination of multiple searches. *NIST Special Publication SP*, pages 243–243.

Gabrilovich, E. and Markovitch, S. (2006). Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, pages 1301–1306.

Giannopoulos, G., Bikakis, N., Dalamagas, T., and Sellis, T. (2010). GoNTogle: A tool for semantic annotation and search. *The Semantic Web: Research and Applications*, pages 376–380.

Große-Bölting, G., Nishioka, C., and Scherp, A. (2015). A comparison of different strategies for automated semantic document annotation. In *Proceedings of the 8th International Conference on Knowledge Capture*, page 8. ACM.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Gulić, M., Vrdoljak, B., and Banek, M. (2016). CroMatcher: An ontology matching system based on automated weighted aggregation and iterative final alignment. *Web Semantics: Science, Services and Agents on the World Wide Web*, 41:50–71.

Hauff, C., Hiemstra, D., and de Jong, F. (2008). A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1419–1420. ACM.

Hawalah, A. and Fasli, M. (2015). Dynamic user profiles for web personalisation. *Expert Systems with Applications*, 42(5):2547–2569.

Hawking, D. (2004). Challenges in enterprise search. In *Proceedings of the 15th Australasian database conference-Volume 27*, pages 15–24. Australian Computer Society, Inc.

Hazman, M., El-Beltagy, S. R., and Rafea, A. (2012). An ontology based approach for automatically annotating document segments. *International Journal of Computer Science*, (v9):i2.

He, B. and Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *International Symposium on String Processing and Information Retrieval*, pages 43–54. Springer.

Hersh, W., Price, S., and Donohoe, L. (2000). Assessing thesaurus-based query expansion using the UMLS metathesaurus. In *Proceedings of the AMIA Symposium*, page 344. American Medical Informatics Association.

Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An Electronic Lexical Database*, 305:305–332.

Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E. G., and Milios, E. (2006). Information retrieval by semantic similarity. *International Journal on Semantic Web and Information Systems*, 2(3):55–73.

Honkela, T. and Hyvarinen, A. (2004). Linguistic feature extraction using independent component analysis. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 1. IEEE.

Huang, M., Névéol, A., and Lu, Z. (2011). Recommending MeSH terms for annotating biomedical articles. *Journal of the American Medical Informatics Association*, 18(5):660–667.

Huber, R. and Klump, J. (2015). Agenames a stratigraphic information harvester and text parser. *Earth Science Informatics*, 8(1):125–134.

Hulpus, I., Hayes, C., Karnstedt, M., and Greene, D. (2013). Unsupervised graph-based topic labelling using DBpedia. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, pages 465–474. ACM.

Husein, I. G., Akbar, S., Sitohang, B., and Azizah, F. N. (2016). Review of ontology matching with background knowledge. In *Data and Software Engineering (ICoDSE), 2016 International Conference on*, pages 1–6. IEEE.

Jain, P., Hitzler, P., Sheth, A. P., Verma, K., and Yeh, P. Z. (2010). Ontology alignment for linked open data. In *The Semantic Web–ISWC 2010*, pages 402–417. Springer.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

Jiang, Y., Wang, X., and Zheng, H.-T. (2014). A semantic similarity measure based on information distance for ontology alignment. *Information Sciences*, 278:76–87.

Katz, G., Shtock, A., Kurland, O., Shapira, B., and Rokach, L. (2014). Wikipedia-based query performance prediction. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 1235–1238. ACM.

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

Knappe, R., Bulskov, H., and Andreasen, T. (2007). Perspectives on ontology-based querying. *International Journal of Intelligent Systems*, 22(7):739–761.

Krovetz, R. (1997). Homonymy and polysemy in information retrieval. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 72–79. Association for Computational Linguistics.

Krovetz, R. and Croft, W. B. (1992). Lexical ambiguity and information retrieval. *ACM Transactions on Information Systems (TOIS)*, 10(2):115–141.

Kwok, K. L. and Chan, M. (1998). Improving two-stage ad-hoc retrieval for short queries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 250–256. ACM.

Lassila, O. and McGuinness, D. (2001). The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, 6(5):2001.

Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, 49(2):265–283.

Leal Bando, L., Scholer, F., and Turpin, A. (2015). Query-biased summary generation assisted by query expansion. *Journal of the Association for Information Science and Technology*, 66(5):961–979.

Lee, J. H. (1997). Analyses of multiple evidence combination. In *ACM SIGIR Forum*, volume 31, pages 267–276. ACM.

Li, J., Tang, J., Li, Y., and Luo, Q. (2009). RiMOM: A dynamic multistrategy ontology alignment framework. *Knowledge and Data Engineering, IEEE Transactions on*, 21(8):1218–1232.

Li, Y., Liu, Z., and Zhu, H. (2014). Enterprise search in the big data era: Recent developments and open challenges. *Proceedings of the VLDB Endowment*, 7(13):1717–1718.

Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., and Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150.

Lin, D. (1998). An information-theoretic definition of similarity. In *ICML*, volume 98, pages 296–304.

Lin, F. and Sandkuhl, K. (2008). A survey of exploiting WordNet in ontology matching. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 341–350. Springer.

Liu, T.-Y. (2011). *Learning to Rank for Information Retrieval*. Springer Science & Business Media.

Lopez, V., Uren, V., Sabou, M. R., and Motta, E. (2009). Cross ontology query answering on the semantic web: An initial evaluation. In *Proceedings of the Fifth International Conference on Knowledge Capture*, pages 17–24. ACM.

Lu, Z., Kim, W., and Wilbur, W. J. (2009). Evaluation of query expansion using MeSH in PubMed. *Information Retrieval*, 12(1):69–80.

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

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Martínez-Romero, M., Vázquez-Naya, J. M., Nóvoa, F. J., Vázquez, G., and Pereira, J. (2013). A genetic algorithms-based approach for optimizing similarity aggregation in ontology matching. In *International Work-Conference on Artificial Neural Networks*, pages 435–444. Springer.

Medelyan, O. (2009). *Human-competitive Automatic Topic Indexing*. PhD thesis, The University of Waikato.

Meij, E., Bron, M., Hollink, L., Huurnink, B., and de Rijke, M. (2011). Mapping queries to the linking open data cloud: A case study using DBpedia. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):418–433.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Monge, A. E., Elkan, C., et al. (1996). The field matching problem: Algorithms and applications. In *KDD*, pages 267–270.

Ngo, D. and Bellahsene, Z. (2012). YAM++: A multi-strategy based approach for ontology matching task. In *Knowledge Engineering and Knowledge Management*, pages 421–425. Springer.

Ngo, D., Bellahsene, Z., and Coletta, R. (2011). A generic approach for combining linguistic and context profile metrics in ontology matching. In *On the Move to Meaningful Internet Systems: OTM 2011*, pages 800–807. Springer.

Ngo, D., Bellahsene, Z., and Todorov, K. (2013). Opening the black box of ontology matching. In *Extended Semantic Web Conference*, pages 16–30. Springer.

Nkisi-Orji, I. (2016). Semantic information retrieval for geoscience resources: results and analysis of an online questionnaire of current web search experiences.

Ohsawa, Y., Benson, N. E., and Yachida, M. (1998). KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on*, pages 12–18. IEEE.

Otero-Cerdeira, L., Rodríguez-Martínez, F. J., and Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. *Expert Systems with Applications*, 42(2):949–971.

Paralic, J. and Kostial, I. (2003). Ontology-based information retrieval. In *Proceedings of the 14th International Conference on Information and Intelligent systems (IIS 2003), Varazdin, Croatia*, pages 23–28.

Paralic, J., Sabol, T., and Mach, M. (2002). A system to support e-democracy. In *International Conference on Electronic Government*, pages 288–291. Springer.

Qian, R. (2013). Understand your world with bing. *Bing search blog, Mar.*

Rada, R. and Bicknell, E. (1989). Ranking documents with a thesaurus. *Journal of the American Society for Information Science*, 40(5):304.

Rada, R., Mili, H., Bicknell, E., and Blettner, M. (1989). Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(1):17–30.

Raskin, R. (2006). Guide to sweet ontologies. *NASA/Jet Propulsion Lab, Pasadena, CA, USA, Available at: http://sweet. jpl. nasa. gov/guide. doc (last accessed: May 2011).*

Reeve, L. and Han, H. (2005). Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM symposium on Applied Computing*, pages 1634–1638. ACM.

Reitsma, F., Laxton, J., Ballard, S., Kuhn, W., and Abdelmoty, A. (2009). Semantics, ontologies and escience for the geosciences. *Computers & Geosciences*, 35(4):706–709.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*.

Richardson, R., Smeaton, A., and Murphy, J. (1994). Using WordNet as a knowledge base for measuring semantic similarity between words.

Rivas, A. R., Iglesias, E. L., and Borrajo, L. (2014). Study of query expansion techniques and their application in the biomedical information retrieval. *The Scientific World Journal*, 2014.

Robertson, S., Zaragoza, H., et al. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Robertson, S. E., Walker, S., Beaulieu, M., Gatford, M., and Payne, A. (1996). Okapi at TREC-4. *NIST Special Publication SP*, pages 73–96.

Ruch, P. (2006). Automatic assignment of biomedical categories: Toward a generic approach. *Bioinformatics*, 22(6):658–664.

Santos, R. L., Macdonald, C., and Ounis, I. (2010). Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th international conference on World wide web*, pages 881–890. ACM.

Shamsfard, M., Nematzadeh, A., and Motiee, S. (2006). ORank: An ontology based system for ranking documents. *International Journal of Computer Science*, 1(3):225–231.

Shtok, A., Kurland, O., and Carmel, D. (2009). Predicting query performance by query-drift estimation. In *Conference on the Theory of Information Retrieval*, pages 305–312. Springer.

Shvaiko, P. and Euzenat, J. (2013). Ontology matching: State of the art and future challenges. *IEEE Transactions on knowledge and data engineering*, 25(1):158–176.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. *Official google blog*, 5.

Spiliopoulos, V., Vouros, G. A., and Karkaletsis, V. (2010). On the discovery of subsumption relations for the alignment of ontologies. *Web Semantics: Science, Services and Agents on the World Wide Web*, 8(1):69–88.

Stoilos, G., Stamou, G., and Kollias, S. (2005). A string metric for ontology alignment. In *International Semantic Web Conference*, pages 624–637. Springer.

Sun, Z., Hu, W., and Li, C. (2017). Cross-lingual entity alignment via joint attribute-preserving embedding. In *International Semantic Web Conference*, pages 628–644. Springer.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. In *Proceedings of the Second International Conference on Information and knowledge Management*, pages 67–74. ACM.

Tatsiopoulos, C. and Boutsinas, B. (2009). Ontology mapping based on association rule mining. In *ICEIS (3)*, pages 33–40.

Trieschnigg, D., Pezik, P., Lee, V., De Jong, F., Kraaij, W., and Rebholz-Schuhmann, D. (2009). MeSH Up: Effective MeSH text classification for improved document retrieval. *Bioinformatics*, 25(11):1412–1418.

Tversky, A. (1977). Features of similarity. *Psychological Review*, 84(4):327.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and 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):14–28.

Uyar, A. and Aliyu, F. M. (2015). Evaluating search features of google knowledge graph and bing satori: entity types, list searches and query interfaces. *Online Information Review*, 39(2):197–213.

Whan Kim, Y. and Kim, J. H. (1990). A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, 46(2):113–136.

Wong, S. K. M., Ziarko, W., Raghavan, V. V., and Wong, P. (1987). On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems (TODS)*, 12(2):299–321.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Xiong, C. and Callan, J. (2015). Query expansion with freebase. In *Proceedings of the 2015 international conference on the theory of information retrieval*, pages 111–120. ACM.

Xiong, C., Merity, S., and Socher, R. (2016). Dynamic memory networks for visual and textual question answering. In *International conference on machine learning*, pages 2397–2406.

Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11. ACM.

Yih, W.-t. and Ma, H. (2016). Question answering with knowledge base, web and beyond. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1219–1221. ACM.

Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342. ACM.

Zhang, Y., Wang, X., Lai, S., He, S., Liu, K., Zhao, J., and Lv, X. (2014). Ontology matching with word embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 34–45. Springer.

Zobel, J. and Moffat, A. (1998). Exploring the similarity space. In *ACM SIGIR Forum*, volume 32, pages 18–34. ACM.

Zouaq, A., Gasevic, D., and Hatala, M. (2012). Voting theory for concept detection. In *Extended Semantic Web Conference*, pages 315–329. Springer.

Appendix A

Publications

- Nkisi-Orji I., Wiratunga N., Massie S., Hui K.Y., and Heaven R., 2018, September. Ontology alignment based on word embedding and random forest classification. In European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases. Springer, Cham
- Nkisi-Orji I., Wiratunga N., Hui K.Y., Heaven R. and Massie S., 2017, September. Taxonomic Corpus-Based Concept Summary Generation for Document Annotation. In International Conference on Theory and Practice of Digital Libraries (pp. 49-60). Springer, Cham
- Nkisi-Orji I., 2016, Semantic information retrieval for geoscience resources: results and analysis of an online questionnaire of current web search experiences. Nottingham, UK, British Geological Survey, 15pp. (OR/16/047)
- Nkisi-Orji I., Wiratunga N., Hui K.Y., Heaven R., and Massie S. Generating taxonomic node descriptors for improved semantic document annotation. Communicated to: International Journal on Digital Libraries (January 2018)
- Nkisi-Orji I., Wiratunga N., Heaven R., Hui K.Y., and Massie S. Ontology-based Model for Selective Search Enhancement. Communicated to: Expert Systems with Applications (December 2018)

Appendix B

Questionnaire on Semantic Search for Geoscience Resources

B.1 Introduction

The questionnaire titled "Semantic web searches for geoscience resources" was completed by staff of British Geological Survey (BGS) in an opt-in manner. The questionnaire was designed to better understand current search habits, preferences, and the reception of semantic search tools.

B.1.1 Response statistics

Responses to this survey were collected online using Survey Monkey[1] between 28 July 2015 and 28 August 2015. Thirty five responses were received over this one-month period.

B.1.2 Questionnaire structure

The questionnaire surveyed two types of search activities and as a result, a number of questions were repeated with respect to each type of search. Section 1 of the questionnaire (Q1-4) relates to search instances where one wants to find a comprehensive list of all relevant results (e.g. literature search, data gathering), so completeness of results is

[1] https://www.surveymonkey.com/

the most important measure of success. Section 2 of the questionnaire (Q5-8) relates to search instances where one is looking for the answer to a specific question, so the relevance ranking of the results is the most important measure of success. Section 3 (Q9-16) posed questions to assess respondents' reception of semantic search features and their preferences in the implementation of such features.

B.2 Questions

B.2.1 Section 1: Literature or data gathering searches

Q1: For this first sort of search, which search applications are useful to you?

- Popular search engine (e.g. Google, Bing, Yahoo)
- Publication citations (e.g. Google Scholar, Science Direct)
- Cross discipline data portal (e.g. data.gov, INSPIRE geoportal, Scottish SDI)
- Earth Sciences catalogue (e.g. NERC Data Catalogue, NERC library, NORA)
- Discipline/community specific catalogue (e.g. MEDIN for marine data, ESDAC for soil data etc)
- BGS intranet tools (dtSearch for text resources, discovery metadata)
- Other (please specify)

Q2: For this first sort of search, how often are you satisfied with the results after:

- Using a small number (<5) of words in a free text search?
- Using a large number (>5) words in a free text search?
- Using logical operators in a free text search (AND/NOT/OR etc)?
- Using advanced search features to search within specific metadata fields (keywords, title, author etc)?

Q3: For this first sort of search, what is the maximum number of search results you are willing to assess, rather than refining your search criteria or changing the search engine?

- 1-10
- 10-20
- 20-50
- 50+

Q4: For this first sort of search, could you give a few examples of some recent searches you conducted, and any comments on the relevance of results returned?

B.2.2 Section 2: Searches that ask specific question

Q5: For this second sort of search, which search applications are useful to you?

- Popular search engine (e.g. Google, Bing, Yahoo)
- Publication citations (e.g. Google Scholar, Science Direct)
- Cross discipline data portal (e.g. data.gov, INSPIRE geoportal, Scottish SDI)
- Earth Sciences catalogue (e.g. NERC Data Catalogue, NERC library, NORA)
- Discipline/community specific catalogue (e.g. MEDIN for marine data, ESDAC for soil data etc)
- BGS intranet tools (dtSearch for text resources, discovery metadata)
- Other (please specify)

Q6: For this second sort of search, how often are you satisfied with the results after:

- Using a small number (<5) of words in a free text search?

- Using a large number (>5) words in a free text search?
- Using logical operators in a free text search (AND/NOT/OR etc)?
- Using advanced search features to search within specific metadata fields (keywords, title, author etc)?

Q7: For this second sort of search, what is the maximum number of search results you are willing to assess, rather than refining your search criteria or changing the search engine?

- 1-10
- 10-20
- 20-50
- 50+

Q8: For this second sort of search, could you give examples of some recent searches you conducted, and any comments on the relevance of results returned?

B.2.3 Section 3: Semantic search features

Q9: How often do you have to perform multiple searches or construct an advanced search query in order to also search all the narrower/child terms of your original search intent?

- always
- usually
- sometimes
- seldom
- never

Q10: How often do you have to perform multiple searches or construct an advanced search query in order to include all the equivalent terms or alternative spellings of your original search intent?

- always
- usually
- sometimes
- seldom
- never

Q11: If a search feature was available that could include the narrower and equivalent terms from controlled vocabularies, would you prefer that this functionality was

- always included implicitly
- included by default but can be turned off by the user
- not included by default but can be turned on by the user
- not included at all, not of benefit to me

Q12: How often do you find that your search results are dominated by results that are not relevant?

- always
- usually
- sometimes
- seldom
- never

Q13: If a search function was available that could search on the intended context/meaning of the search term entered, rather than just matching the term as typed, would you prefer to

- always specify the context/meaning of your search terms as you build the search (e.g. pick them from a controlled vocabulary)
- specify the context/meaning of your search terms only if there is ambiguity (e.g. pick the correct definition from a list)
- let the search engine decide which context/meaning to use, depending on my previous actions or preferences
- not have this feature, not of benefit to me

Q14: Which vocabularies would be useful to you in the sort of semantic search functionality described above?

Q15: Might you be willing to volunteer 1 hour of your time to help evaluate a search tool which implements features like the above?

Q16: Please provide any other relevant comments such as current search challenges, features you value in a search engine (existing or desired), preferred search engines not mentioned in questionnaire etc. mentioned in questionnaire, etc.

Appendix C

Comparison of alternative semantic relatedness approaches for semantic document ranking

C.1 Introduction

The following features are used to describe selected semantic relatedness approaches.

R: Root node of ontology.

$n(x)$: Set of nodes that are upward reachable from x (x inclusive).

$l(x, y)$: Number of nodes in the shortest path between x and y.

$mscs(x, y)$: Most specific common subsumer of x and y. This is the most specific concept shared by x and y.

$P(x)$: Probability of occurrence of x in a corpus.

C.1.1 Wu and Palmer

Semantic relatedness using the Wu and Palmer (1994) algorithm.

$$relatedness(x,y) = \frac{2 * l(mscs(x,y),R)}{l(x,mscs(x,y)) + l(y,mscs(x,y)) + 2 * l(mscs(x,y),R)} \quad \text{(C.1)}$$

C.1.2 Knappe

Semantic relatedness using Knappe et al. (2007) algorithm.

$$relatedness(x,y) = \rho\frac{|n(x) \cap n(y)|}{|n(x)|} + (1-\rho)\frac{|n(x) \cap n(y)|}{|n(y)|} \quad \text{(C.2)}$$

$\rho \in [0,1]$ determines the weight of each sub-expression in equation C.2. We set $\rho = 0.8$ as this gives the algorithm the desired properties with respect to generalisation and specialisation costs and was used by the authors.

C.1.3 Lin

Semantic relatedness using (Lin, 1998) algorithm.

$$relatedness(x,y) = \frac{2 * logP(mscs(x,y))}{logP(x) + logP(y)} \quad \text{(C.3)}$$

$$P(c) = \frac{freq(c)}{N} \quad \text{(C.4)}$$

$freq(c)$ is the frequency of concept c and that of concepts it subsumes in the corpus. N is the total number of concepts in the corpus.

C.2 Result of comparison

- $QESemWUP$: Query expansion (QE) followed by semantic re-rank using Wu and Palmer to measure semantic relatedness between concepts.
- $SemWUP$: Semantic re-rank using Wu and Palmer.
- $SemKNP$: Semantic re-rank using Knappe.

- $SemLIN$: Semantic re-rank using Lin.

Table C.1: Average precision on individual queries in TREC 2006 Genomics track collection

Topic	VSM_{TFIDF}	QE	$QESemWUP$	$SemWUP$	$SemKNP$	$SemLIN$	Query concepts
160	0.137	0.364	**0.365**	0.162	0.164	0.161	1
161	**0.304**	0.012	0.011	0.274	0.274	0.274	1
162	0.002	0.002	0.002	0.002	0.002	0.002	0
163	0.365	0.304	0.304	**0.369**	0.368	0.367	2
165	0.289	0.116	0.177	0.335	0.345	**0.355**	3
166	0.281	0.083	0.075	**0.309**	0.303	0.302	2
167	**0.293**	0.003	0.003	0.280	0.277	0.278	4
168	0.007	0.007	0.007	0.007	0.007	0.007	0
169	0.318	0.284	0.291	**0.335**	0.334	**0.335**	2
170	**0.400**	0.333	**0.400**	**0.400**	**0.400**	**0.400**	1
171	0.003	0.003	0.003	0.003	0.003	0.003	4
172	0.006	0.004	0.005	**0.007**	**0.007**	**0.007**	1
174	0.016	0.016	0.016	0.016	0.016	0.016	0
175	0.247	0.280	**0.306**	0.269	0.270	0.261	2
176	**0.026**	0.023	0.020	0.021	0.021	0.018	1
178	0.056	0.008	0.008	0.060	0.062	**0.064**	3
179	0.069	0.079	**0.084**	0.083	**0.084**	**0.084**	1
181	**0.405**	0.315	0.301	0.404	0.403	0.404	3
182	0.208	0.204	0.204	**0.221**	**0.221**	0.219	3
183	0.170	0.162	0.134	**0.198**	0.195	0.174	2
184	0.019	0.026	**0.167**	**0.167**	**0.167**	**0.167**	4
185	0.379	0.306	0.330	0.431	0.434	**0.440**	3
186	0.331	0.137	0.151	**0.342**	**0.342**	0.341	3
187	0.373	**0.378**	0.355	0.327	0.327	0.327	4

Table C.2: Average precision on individual queries in TREC 2007 Genomics track collection

Topic	VSM_{TFIDF}	QE	$QESemWUP$	$SemWUP$	$SemKNP$	$SemLIN$	Query concepts
200	0.189	0.167	0.175	**0.206**	**0.206**	0.203	4
201	0.093	0.054	0.073	0.116	0.116	**0.118**	2
202	**0.090**	0.068	0.058	0.064	0.063	0.063	2
203	0.406	0.394	0.393	**0.409**	0.408	0.408	3
204	0.555	0.531	0.533	**0.563**	**0.563**	0.562	3
205	0.114	0.054	0.062	**0.130**	0.127	**0.130**	2
206	**0.440**	0.435	0.425	0.429	0.428	0.430	1
207	0.044	0.044	0.044	0.044	0.044	0.044	0
208	**0.263**	0.235	0.181	0.216	0.216	0.214	1
209	0.323	0.323	0.323	0.323	0.323	0.323	0
210	0.016	0.016	0.016	0.016	0.016	0.016	0
211	**0.358**	**0.358**	0.338	0.338	0.340	0.338	2
212	0.255	0.249	0.246	**0.272**	0.270	0.269	2
213	0.307	0.124	0.118	0.312	**0.313**	0.308	2
214	0.297	0.170	0.204	0.312	**0.314**	0.308	2
215	0.275	0.065	0.078	**0.301**	0.299	0.300	3
216	0.035	0.014	0.022	0.040	**0.041**	0.040	3
217	**0.002**	0.000	0.000	0.001	**0.002**	0.001	3
218	0.253	0.005	0.006	0.262	0.261	**0.264**	2
219	0.057	0.054	**0.077**	0.073	0.075	0.074	3
220	**0.777**	0.729	0.675	0.653	0.665	0.704	1
222	0.081	0.081	0.081	0.081	0.081	0.081	0
223	0.077	0.148	**0.152**	0.091	0.093	0.096	1
224	0.028	0.016	0.013	0.033	**0.035**	0.027	3
225	0.067	0.067	0.067	0.067	0.067	0.067	0
226	0.251	0.427	**0.432**	0.210	0.211	0.213	2
227	0.106	0.049	0.037	0.106	0.106	**0.109**	2
228	0.005	**0.006**	0.005	0.004	0.004	0.004	4
229	**0.480**	0.203	0.194	0.455	0.455	0.451	2
230	0.086	**0.117**	0.085	0.079	0.079	0.082	1
231	0.040	0.008	0.007	0.049	**0.056**	**0.056**	2
232	0.067	**0.125**	0.106	0.064	0.064	0.064	1
233	0.040	0.001	0.001	**0.070**	**0.070**	**0.070**	3
234	0.048	**0.059**	**0.059**	0.050	0.051	0.051	2
235	0.145	0.033	0.032	**0.146**	0.143	**0.146**	2