Ontology-based Information Retrieval

Henrik Bulskov Styltsvig

A dissertation Presented to the Faculties of Roskilde University in
Partial Fulfillment of the Requirement for the
Degree of Doctor of Philosophy

Computer Science Section
Roskilde University, Denmark
May 2006

Abstract

In this thesis, we will present methods for introducing ontologies in information retrieval. The main hypothesis is that the inclusion of conceptual knowledge such as ontologies in the information retrieval process can contribute to the solution of major problems currently found in information retrieval.

This utilization of ontologies has a number of challenges. Our focus is on the use of similarity measures derived from the knowledge about relations between concepts in ontologies, the recognition of semantic information in texts and the mapping of this knowledge into the ontologies in use, as well as how to fuse together the ideas of ontological similarity and ontological indexing into a realistic information retrieval scenario.

To achieve the recognition of semantic knowledge in a text, shallow natural language processing is used during indexing that reveals knowledge to the level of noun phrases. Furthermore, we briefly cover the identification of semantic relations inside and between noun phrases, as well as discuss which kind of problems are caused by an increase in compoundness with respect to the structure of concepts in the evaluation of queries.

Measuring similarity between concepts based on distances in the structure of the ontology is discussed. In addition, a shared nodes measure is introduced and, based on a set of intuitive similarity properties, compared to a number of different measures. In this comparison the shared nodes measure appears to be superior, though more computationally complex. Some of the major problems of shared nodes which relate to the way relations differ with respect to the degree they bring the concepts they connect closer are discussed. A generalized measure called weighted shared nodes is introduced to deal with these problems.

Finally, the utilization of concept similarity in query evaluation is discussed. A semantic expansion approach that incorporates concept similarity is introduced and a generalized fuzzy set retrieval model that applies expansion during query evaluation is presented. While not commonly used in present information retrieval systems, it appears that the fuzzy set model comprises the flexibility needed when generalizing to an ontology-based retrieval model and, with the introduction of a hierarchical fuzzy aggregation principle, compound concepts can be handled in a straightforward and natural manner.

Resumé (in danish)

Fokus i denne afhandling er anvendelse af ontologier i informationssøgning (Information Retrieval). Den overordnede hypotese er, at indføring af konceptuel viden, så som ontologier, i forbindelse med forespørgselsevaluering kan bidrage til løsning af væsentlige problemer i eksisterende metoder.

Denne inddragelse af ontologier indeholder en række væsentlige udfordringer. Vi har valgt at fokusere på similaritetsmål der baserer sig på viden om relationer mellem begreber, på genkendelse af semantisk viden i tekst og på hvordan ontologibaserede similaritetsmål og semantisk indeksering kan forenes i en realistisk tilgang til informationssøgning.

Genkendelse af semantisk viden i tekst udføres ved hjælp af en simpel natursprogsbehandling i indekseringsprocessen, med det formål at afdække substantivfraser. Endvidere, vil vi skitsere problemstillinger forbundet med at identificere hvilke semantiske relationer simple substantivfraser er opbygget af og diskutere hvordan en forøgelse af sammenføjning af begreber influerer på forespørgselsevalueringen.

Der redegøres for hvorledes et mål for similaritet kan baseres på afstand i ontologiers struktur, og introduceres et nyt afstandsmål – “shared nodes”. Dette mål sammenlignes med en række andre mål ved hjælp af en samling af intuitive egenskaber for similaritetsmål. Denne sammenligning viser at “shared nodes” har fortrin frem for øvrige mål, men også at det er beregningsmæssigt mere indviklet. Der redegøres endvidere for en række væsentlige problemer forbundet med “shared nodes”, som er relateret til den forskel der er mellem relationer med hensyn til i hvor høj grad de bringer de begreber de forbinder, sammen. Et mere generelt mål, “weighted shared nodes”, introduceres som løsning på disse problemer.

Afslutningsvist fokuseres der på hvorledes et similaritetsmål, der sammenligner begreber, kan inddrages i forespørgselsevalueringen. Den løsning vi præsenterer indfører en semantisk ekspansion baseret på similaritetsmål. Evalueringsmetoden der anvendes er en generaliseret “fuzzy set retrieval” model, der inkluderer ekspansion af forespørgsler. Selvom det ikke er almindeligt at anvende fuzzy set modellen i informationssøgning, viser det sig at den har den fornødne fleksibilitet til en generalisering til ontologibaseret forespørgselsevaluering, og at indførelsen af et hierarkisk aggregeringsprincip giver mulighed for at behandle sammensatte begreber på en simpel og naturlig måde.

Acknowledgements

The work presented here has been completed within the framework of the interdisciplinary research project OntoQuery and the Design and Management of Information Technology Ph.D. program at Roskilde University. First of all, I would like to thank my supervisor Troels Andreasen for constructive, concise and encouraging supervision from beginning to end. His advice has been extremely valuable throughout the course of the thesis. I would furthermore like to thank my friend and colleague Rasmus Knappe for fruitful discussions and support throughout the process.

My thanks also goes to the members of the OntoQuery project, Jørgen Fischer Nilsson, Per Anker Jensen, Bodil Nistrup Madsen, and Hanne Erdman Thomsen, as well as the other Ph.D. students associated with the project.

Finally, I would like to thank my friends and family for putting up with me during a somewhat stressful time, and I would especially like to thank my wife Kirsten for standing by me and giving me the room to make this happen.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the years, the volume of information available through the World Wide Web has been increasing continuously, and never has so much information been so readily available and shared among so many people. The role of searching applications has therefore changed radically from systems designed for special purposes with a well-defined target group to general systems for almost everyone. Unfortunately, the unstructured nature and huge volume of information accessible over networks have made it increasingly difficult for users to sift through and find relevant information. Numerous information retrieval techniques have been developed to help deal with this problem.

The information retrieval techniques commonly used are based on keywords. These techniques use keyword lists to describe the content of information, but one problem with such lists is that they do not say anything about the semantic relationships between keywords, nor do they take into account the meaning of words and phrases.

It is often difficult for ordinary users to use information retrieval systems based on these commonly used keyword-based techniques. Users frequently have problems expressing their information needs and translating those needs into requests. This is partly because information needs cannot be expressed appropriately in the terms used by the system, and partly because it is not unusual for users to apply search terms that are different from the keywords information systems use. Various methods have been proposed to help users choose search terms and formulate requests. One widely used approach is to incorporate a thesaurus-like component into the information system that represents the important concepts in a particular subject area as well as the semantic relationships connecting them.

Using conceptual knowledge to help users formulate their requests is just one method of introducing conceptual knowledge to information retrieval. Another is to use conceptual knowledge as an intrinsic feature of the system in the process of retrieving the information. Obviously, such methods are not mutually exclusive and would complement one another well.

In the late nineties, Tim Berners-Lee [1998] introduced the idea of a Semantic Web, where machine readable semantic knowledge is attached to all information. The semantic knowledge attached to information is united by means of ontologies, i.e. the concepts attached to information are mapped into these ontologies. Machines can then determine that concepts in different pieces of information are actually the same due to the position they hold in the ontologies. The representation of the semantic knowledge in the Semantic Web is done by use of a special markup language, as the focus is the hypertext structure of the World Wide Web. In information retrieval systems, this representation serves as the fundamental description of the information captured by the system, and thus the representation requires completely different structures. Nevertheless, the mapping of concepts in information into conceptual models, i.e. ontologies, appears to be a useful method for moving from keyword-based to concept-based information retrieval.

Ontologies can be general or domain specific, they can be created manually or automatically, and they can differ in their forms of representation and ways of constructing relationships between the concepts, but they all serve as an explicit specification of a conceptualization. Thus, if the users of a given information retrieval system can agree upon the conceptualization in the ontologies used, then the retrieval process can benefit from using these in the evaluation and articulation of requests.

In this thesis, we will present methods for introducing ontologies in information retrieval along this line. The main hypothesis is that the inclusion of conceptual knowledge in the information retrieval process can contribute to the solution of major problems in current information retrieval systems. When working with ontology-based information retrieval systems, one important aim is to utilize knowledge from a domain-specific ontology to obtain better and more exact answers on a semantic basis, i.e. to compare concepts rather than words. In this context, better answers are primarily more fine-grained and better-ranked information base objects that are obtained by exploiting improved methods for computing the similarity between a query and the objects from the information base. The idea is to use measures of similarity between concepts derived from the structure of the ontology, and by doing so, replace reasoning over the ontology with numerical similarity computation.

The benefit of applying ontology-based similarity measures is dependant on the ability of the semantic interpretation of queries and information objects, as the aim is to relate information found in these to the senses covered by the ontologies. In information retrieval, the interpretation process is referred to as indexing. During this process, descriptions of queries and information objects are extracted, and either stored (the analysis of the information base) or used for searching for similar information. It is therefore required to introduce natural language processing in the indexing process in order to extract the intrinsic semantic knowledge of the (textual) information. Natural language processing is challenging, and completely exposing the semantics is very demanding, not to say impossible. We, therefore, present ways to achieve a shallow processing that recognizes noun phrases, which can then be mapped into the ontology to create the semantic foundation of the representation used.

The central topics of this thesis can thus be summarized as measures of semantic similarity and semantic information analysis, denoted "ontological similarity" and "ontological indexing", respectively, and methods for uniting these into a realistic information retrieval scenario, and in so doing, promote semantics in the information retrieval process. To achieve this, a generalized fuzzy set retrieval model with an ontology-based query expansion is used. Finally, a number of the techniques presented are used in a prototype system intended to be the foundation for the evaluation and testing of the presented methodologies.

1.1 Research Question

The main objective in this thesis is to provide realistically efficient means to move from keyword-based to concept-based information retrieval utilizing ontologies as reference for conceptual definitions. More specifically the key research question is the following. How do we recognize concepts in information objects and queries, represent these in the information retrieval system, and use the knowledge about relations between concepts captured by ontologies in the querying process?

The main aspects in focus in this thesis are the following:

1. recognition and mapping of information in documents and queries into the ontologies,

2. improvement of the retrieval process by use of similarity measures derived from knowledge about relations between concepts in ontologies, and

3. how to weld the ideas of such ontological indexing and ontological similarity into a realistic information retrieval scenario

The first part of the research question concerns extraction of semantic knowledge from texts, construction of (compound) concepts in a lattice-algebraic representation, and a word sense disambiguation of this knowledge such that it can be mapped into the ontology in use. The second part concerns the development of scalable similarity measures, where the idea is to compare concepts on behalf of the structure in the ontology, and the hypothesis that measures which incorporates as many aspects as possible will be closer to rendering human similarity judgments. Finally, the object of the last part is how to bring the two first parts into play in information retrieval systems. This is completed by use of a query expansion technique based on concept similarity measures and a flexible retrieval model with the achievability of capturing the paradigm shift of representations from simple collections of words to semantic descriptions.

1.2 Thesis Outline

Chapter 2, which presents classical information retrieval models, including the fuzzy set retrieval model, discusses the strengths and weaknesses of the models. In addition, retrieval evaluation is briefly covered. Chapter 3 focuses on ontologies. A formal definition of ontology and a number of different formalisms for the representation of ontologies are presented. This chapter also presents and discusses a number of ontological resources, with a focus on WordNet, a large lexical database for English. The topics examined in Chapter 4 are all related to what is denoted as ontological indexing. The discussion begins with a representation of information in general and a representation of semantic knowledge in particular. Next, the extraction of semantic knowledge aimed at a (rather ad hoc) method that can produce semantical representations from information is covered. Finally, the notion of instantiated ontologies is presented. Chapter 5 then starts with a definition of a set of intuitive, qualitative properties used throughout the presentation of a number of different similarity measures to discuss their strengths and weaknesses. Special emphasis is put on the weighted shared node measure. At the end of this chapter, a simple experiment is presented in which the weighted shared node similarity measure is compared to a number of the other presented measures through comparison to human similarity judgments. In Chapter 6, query evaluation is covered. We introduce a technique called semantic expansion to incorporate a similarity measure in the query evaluation. Moreover, the generalized version of the fuzzy retrieval model is presented where order weighted averaging and hierarchical aggregation are included. A prototype system that unites some of the introduced techniques is presented and discussed in Chapter 7. Finally, we conclude and indicate perspectives that can be used as the subject of further work.

1.3 Foundations and Contributions

This dissertation is funded in part by the OntoQuery research project (Ontology-based Querying)[Andreasen *et al.*, 2000; Andreasen *et al.*, 2002; Onto-

Query, 2005][1] and in part by Roskilde University. Research issues and the results presented in this dissertation are thus related to and inspired by work done in the OntoQuery project.

The main contributions covered in this thesis concern the design of measures of similarity based on the structure of ontologies, a discussion of representation and the extraction of descriptions, methods composed to support ontological query evaluation, and a discussion of the challenges in uniting it all into a prototype that can serve as the foundation for evaluation and testing.

While working with this thesis a number of papers have been submitted to different conferences and journals. These papers cover the progression in the design and application of ontological similarity measures, along with a number of closely related topics. The following is a short list of some of the primary research topics covered by these papers and included in this thesis:

- **Similarity measures**, with the measures Shared Nodes and Weighted Shared Nodes as the final contributions.
- **Instantiated Ontologies**, a ontology modeling method usend to restrict a general ontology to concepts that occur in, for instance, a document collection or a query.
- **Querying evaluation**, with Hierarchical Ordered Weighted Averaging Aggregation as the primary contribution to simplify query evaluation by comparing descriptions of the query with the descriptions of the documents.

The listed research topics are collaborative work, but my contribution is central and thus these topics are included and have a central role in this thesis.

[1] The following institutions have participated in this project: Centre for Language Technology, The Technical University of Denmark - Informatics and Mathematical Modelling, Copenhagen Business School - Computational Linguistics, Roskilde University - Intelligent Systems Laboratory and the University of Southern Denmark.

Chapter 2

Information Retrieval

Information retrieval deals with access to information as well as its representation, storage and organization. The overall goal of an information retrieval process is to retrieve the information relevant to a given request. The criteria for complete success is the retrieval of all the relevant information items stored in a given system, and the rejection of all the non-relevant ones. In practice, the results of a given request usually contain both a subset of all the relevant items, plus a subset of irrelevant items, but the aim remains, of course, to meet the ideal criteria for success.

In theory, it is possible for information of any kind to be the object of information retrieval. One information retrieval system could handle different kinds of information simultaneously, e.g. information retrieval in multi-media environments. However, in the majority of information retrieval systems, the items handled are *text documents*, and information retrieval is therefore often regarded as synonymous with *document retrieval* or *text retrieval*. The notion of text documents in information retrieval incorporates all types of texts, spanning from complete texts, such as articles, books, web pages, etc., to minor fragments of text, such as sections, paragraphs, sentences, etc. A given system, for example, could be designed for one specific type of document, or heterogeneous documents, and be either monolingual or multilingual.

Information retrieval systems do not actually retrieve information, but rather documents from which the information can be obtained if they are read and understood [Van Rijsbergen, 1979; Lancaster, 1979]. To be more precise, that which is being retrieved is the system's internal description of the documents, thus the process of fetching the documents being represented is a separate process [Lancaster, 1979]. Despite this loose definition, *information retrieval* is the term commonly used to refer to this kind of system, and thus, whenever the term *information retrieval* is used, it refers to this *text-document-description retrieval* definition.

In Figure 2.1 [Ingwersen, 1992], which shows the basic concepts of an infor-

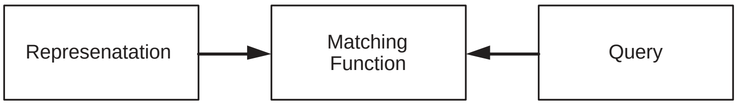


Figure 2.1: *A simple model for information retrieval*

mation retrieval system, *representation* is defined as the stored information, *matching function* as a certain search strategy for finding the stored information and *queries* as the requests to the system for certain specific information.

Representation comprises an abstract description of the documents in the system. Nowadays, more and more documents are *full text documents*, whereas previously, representation was usually built on *references to documents* rather than the documents themselves, similar to bibliographic records in library systems [Jones and Willett, 1997]. References to documents are normally semi-structured information with predefined slots for different kinds of information, e.g. *title*, *abstract*, *classification*, etc., while full text documents typically are unstructured, except for the syntax of the natural language.

The matching function in an information retrieval system models the system's notion of similarity between documents and queries, hence defining how to compare requests to the stored descriptions in the representation. As discussed later in this chapter, each model has its advantages and disadvantages, with no single strategy being superior to the others.

In Figure 2.2, a more detailed view of the major functions in most information retrieval systems is shown [Lancaster, 1979]. At the top of the figure, the input side of the system is defined as a set of selected documents. These documents are organized and controlled by the indexing process, which is divided into two parts, a conceptual analysis and a transformation. The conceptual analysis, or content analysis, recognizes the content of the documents and the transformation transforms this information into the internal representation. This representation is stored and organized in a database for later (fast) retrieval. The bottom of the figure defines the output side, which is very similar to the input side. The conceptual analysis and transformation of requests recognizes the semantics and transforms this knowledge into a representation similar to the representation used in the indexing process. Then, some type of search strategy (matching function) is used to retrieve documents by comparing the description of the request with the descriptions of the documents. If the request process is iterative, as shown in the figure by broken lines, it uses either documents or document descriptions in the process of obtaining the information needed. This iterative querying process is called *query reformulation*.

In traditional information retrieval systems, the typical description of documents is as a collection of words, ranging from all the words contained in the

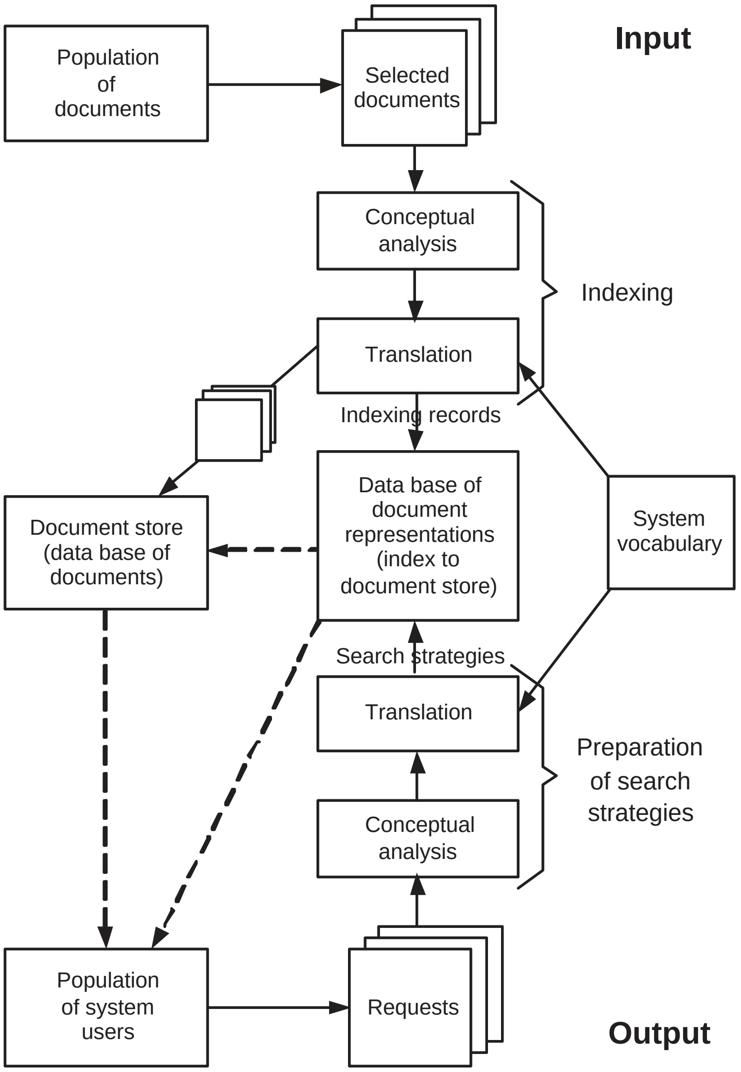


Figure 2.2: *The major functions performed in many types of information retrieval systems*

documents to a smaller collection of keywords chosen to point out the content of the documents, that are not necessarily found in the particular document.

In Figure 2.2, *system vocabulary* refers to a collection of words valid for use as keywords in the indexing and querying process. If a system's vocabulary equals the collection of all words used in the stored documents, then no restrictions are defined, otherwise, the vocabulary is used to control the system's collection of valid keywords. More compound pieces of information, like multi-word fragments of text, natural language phrases, etc., can also be used as parts of descriptions, but would obviously require special kinds of representation and processing techniques.

In data retrieval the aim is to retrieve all the data that satisfies a given query. If the database is consistent, the result would be exactly all the data that can possibly be found in the domain covered by the database. One single erroneous object means total failure. The query language in data retrieval clearly defines the conditions for retrieving the data in the well-defined structure and semantics of the database. Knowledge of the querying language and the structure and semantics of the database is presupposed for querying a database. In information retrieval, however, some of the documents retrieved might be inaccurate and erroneous without disqualifying the result in general. The descriptions of both documents and requests are a kind of "interpretation" of content. Consequently, the documents retrieved are not a strict match based on well-defined syntax and semantics, but the estimated "relevant" information. The notion of *relevance* is the core of information retrieval, and the retrieved documents are those found to be most relevant to the given request under the conditions available (representation, search strategy, and query).

The notion of *descriptions* and the *indexing process* will be discussed further in Chapter 4. Before going into detail on specific search strategies, a short introduction to *term weighting* is given. Following the section on search strategies is a discussion about retrieval evaluation, and finally, a summary of the chapter.

2.1 Search Strategies

The motivation for entering requests into an information retrieval system is an *information need* [Lancaster, 1979], and the success of a given retrieval system depends on the system's capability to provide the user with the information needed within a reasonable time and with a straightforward interface for posing requests and collecting the results[Austin, 2001][1].

Until the mid-nineties, operational information retrieval systems were characterized, almost without exception, by their adoption of the *Boolean model*[2] of searching. A surprising strategy considering the antagonism between the intrinsic restrictiveness of the Boolean model and the intrinsic uncertainty in information retrieval, but an understandable one, on the other hand, when the constraint on the file-handling technology and the presumption that users with access to the systems were professionals are considered [Jones and Willett, 1997]. The intrinsic uncertainty of information retrieval is caused by the fact that the representation of documents is "uncertain", i.e. the extraction of

[1] The out of context interpretation of Mooers' Law.

[2] Described in Section 2.1.2.

information from documents or queries is a highly uncertain process [Crestani *et al.*, 1998].

In the majority of the information systems available, the interface for posing queries is a collection of words, which is a rather limited language for expressing an information need. The first step for a user in the process of retrieving information is therefore a transformation of their representation of the information needed into a collection of words. This is by no means a trivial process and requires some training to master, even at a novice level, but since this is the kind of interface most often used, training opportunities are abundant. The drawback is that it is very difficult to introduce other interfaces, such as natural language, since users (well-trained) are accustomed to this limited interface. Nevertheless, independent of the query language, a transformation is needed for the user to express their information need, with some prior knowledge about the system preferable.

The information retrieval system can provide help during the *query definition process* in many ways, partly via interaction with the user and partly via refinements on the evaluation process. Interaction with the user can take place either prior to posing the query or as an iterative query reformulation process. Access to knowledge about the representation of documents, for instance, browsing the set of keywords, is an example of prior interaction, while *relevance feedback* is an example of iterative interaction. Relevance feedback is a technique in which the user points out relevant documents from the result of a (tentative) query. The system then uses this subset of documents in a query reformulation for refinement of the query [Salton and McGill, 1997]. This process can be repeated until a satisfactory result is found.

Two examples of a refinement of the evaluation process are softening of the evaluation criteria and *query expansion*. One example of the former is replacement of *strict match* with *best match*, a partial match where also documents partially fulfilling the query are accepted. The retrieved documents are then ordered based on their partial fulfillment of the queries, with the ones closest to the queries first, hence the name *best match*. This gives the user the possibility of adding more words to the query without the risk of getting an empty result because some words in the query are mutually exclusive, since no documents are represented by that combination of words. In query expansion, the system expands the query with additional information, for instance, synonyms. This expansion is done after the query is posed to the system, but before the evaluation process.

Two assumptions common for the retrieval models presented here are the *document independence assumption* and the *term independence assumption*. The document independence assumption [Robertson, 1997] states that the relevance of one document with respect to a given query is independent of other retrieved documents in the same query. As a counterargument to the document independence assumption, one could argue that the usefulness of retrieved documents is dependent on the other retrieved documents, e.g. after retrieving one document (and having read it) the next document similar to that one would be less useful, but the relatedness would still be the same. Similarly, the term independence assumption states with regard to terms that they are independent of other terms in the same document. Since concepts in documents are often semantically related to one another, the term independence is an unrealistic assumption. The reason for defining the term independence assumption is that it leads to models that are easy to implement, extend and conceptualize.

2.1.1 Term Weighting

One of the simplest representations of documents in information retrieval is a collection of terms corresponding to all the words contained in the documents. The problem with such representation is that not all words in a document specify the content equally, therefore the indexing process should provide some kind of differentiation among the words present. The classical approach for doing this is called *term weighting*. Weights indicate relevance and a simple and well-known weighting principle is to derive weights from the frequency with which words appear in the documents. The justification for the use of the frequency of words is that words occur in natural language unevenly, and classes of words are distinguishable by the frequency of their occurrence [Luhn, 1958]. Actually, the occurrence characteristics of words can be characterized by the constant rank-frequency law of Zipf:

$$\text{frequency} \cdot \text{rank} \simeq \text{constant} \tag{2.1}$$

where *frequency* is the number of occurrences of a given word and *rank* is achieved by sorting the words by *frequency* in decreasing order. Thus, the frequency of a given word multiplied by the rank of the word is approximately equal to the frequency of another word multiplied by its rank.

A simple approach to term weighting is *term frequency* $tf_{i,j}$, where each term t_i is weighted according to the number of occurrences of the word associated with the term in document d_j. The selection of significant words is then defined by thresholds rejecting the most frequent and the least frequent words. The reason for rejecting the high and low frequency words is that neither of them are good content identifiers. This idea entails the notion of *resolving power*, that is the degree of discrimination for a given word. The limits are illustrated in Figure 2.3. [Luhn, 1958].

One of the problems with this simple approach is that some words may be assigned to many of the items in a collection and yet still be more relevant in

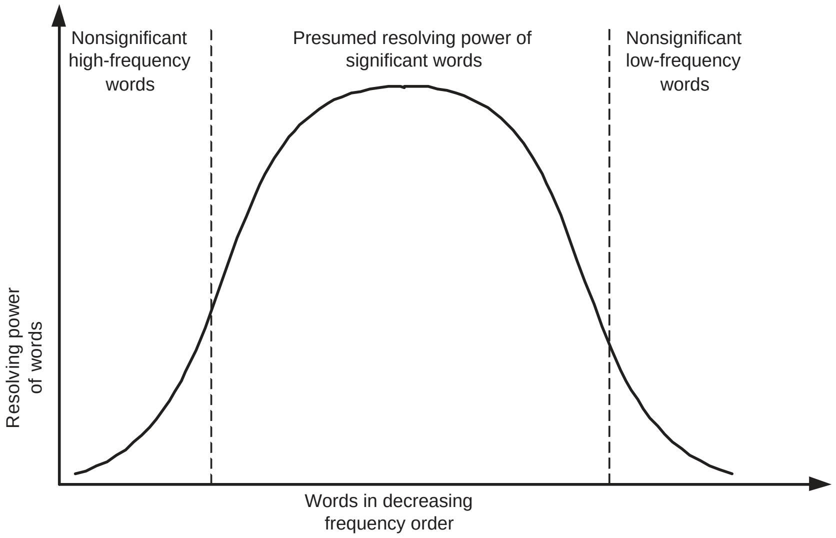


Figure 2.3: *Resolving power of significant words*

some items than in others. Such words could be rejected by a high-frequency threshold, which means that a *relative frequency measure* would be more appropriate. A relative frequency measure can be divided into a local and a global weighting. The local weight for a given document defines the relevance of the terms appearing, for instance, $tf_{i,j}$ or in a normalized version:

$$f_{i,j} = \frac{tf_{i,j}}{\max_{1 \leq l \leq n_j}(tf_{l,j})}$$

where n_j is the number of terms present in the document d_j [Baeza-Yates and Ribeiro-Neto, 1999].

One well-known global weight is the *inverse document frequency*, which assigns the level of discrimination to each word in a collection of items (documents). A word appearing in most items should have a lower global weight than words appearing in few items [Salton and McGill, 1986] , such that:

$$idf_i = \log \frac{N}{n_i} \tag{2.2}$$

where n_i are the number of items in which term t_i appear, and N is the total number of documents in the collection. One of the most important relative frequency measures is given by:

$$w_{i,j} = f_{ik} \times log \frac{N}{n_i} \tag{2.3}$$

which assigns a weight to each word in a document depending not only on the local frequency of the word in the item, but also on the resolving power of that word in the collection of documents [Salton and McGill, 1986]. This approach is known as the *tf-idf* (term frequency - inverted document frequency).

The approaches described here constitute the old central ideas about term weighting. Over the last couple of decades, many alternative approaches and modifications on the ones given here have been developed. However, the common goal for all these approaches is to assign a weight to each word in each item, indicating to what degree the word describes the content of the document.

2.1.2 Boolean Model

As mentioned above, the Boolean model was the overall dominating search strategy in information retrieval until the mid-nineties. The primary reason for this was, on the one hand, that the model is simple and intuitive, and on the other, that there was no clear consensus among researchers as to which of the many non-Boolean models was the best [Cooper, 1997], and thus there was no immediate alternative.

The *Boolean model* is based on set theory and Boolean algebra. Queries specified as Boolean expressions have precise semantics in addition to being relatively intuitive, at least with regard to simple queries. The retrieval strategy in the Boolean model is based on the *binary decision criterion*, which denotes that a document is predicted either to be relevant or not-relevant to a given query, hence there are no intermediate relevance judgments.

In the Boolean model documents are usually represented as binary vectors of *index terms*, specifying which terms are assigned to the documents. Queries are lists of keywords combined with the Boolean operators, AND, OR and NOT ($\wedge, \vee, \neg$), thus forming Boolean expressions. A typical strategy is a conjunctive reading of the different aspects or *facets* in the queries and a disjunctive reading of terms in the facets, e.g. synonyms, inflections, etc., for example:

$$q = (\text{“car”} \vee \text{“auto”} \vee \text{“automobile”}) \wedge (\text{“holiday”} \vee \text{“vacation”})$$

for a query on “car vacations”. In more complicated queries, precedence often has to be explicated by parentheses, e.g. in query $[a \wedge b \vee c]$, it is necessary to be explicit and choose either query $[(a \wedge b) \vee c]$ or $[a \wedge (b \vee c)]$. This is often too difficult to handle for normal users, and most queries are therefore very simple. Even the very simple expressions, however, have interpretations which create confusion for novice users. One example of this is the confusion between the Boolean “AND” and “OR” for English speakers because in ordinary conversations a noun phrase of the form “A AND B” usually refers to more entities than “A” alone, while in logic, the use of “AND” refers to fewer documents than “A” alone [Cooper, 1997].

Boolean expressions can be represented as a disjunction of conjunctions referred to as the *disjunctive normal form* (DNF). For instance, the query $q = t_a \wedge (t_b \vee \neg t_c)$ can be represented in disjunctive normal form as:

$$q_{DNF} = (t_a \wedge t_b) \vee (t_a \wedge \neg t_c)$$

This corresponds to the vector representation $(1,1,1) \vee (1,1,0) \vee (1,0,0)$, where each of the components are a binary vector associated with the tuple (t_a, t_b, t_c), as shown in Figure 2.4 [Baeza-Yates and Ribeiro-Neto, 1999].

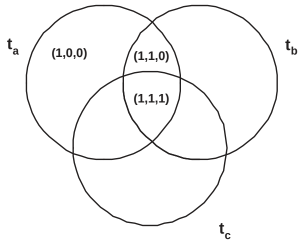


Figure 2.4: *The three conjunctive components of the query $q = t_a \wedge (t_b \vee \neg t_c)$*

A major drawback of the Boolean model is the exact match caused by the binary decision criterion. One way to remedy this is to model the likelihood of relevance as the number of index terms that the query and a document have in common. This presupposes that both the query and the documents are represented as sets of index terms. A *best match* evaluation can then be performed where the retrieved documents are ordered by decreasing number of elements in common with the query, thus listing the most relevant documents first. The strategy for posing queries would then be completely different, not requiring considerations about whether to combine terms with “AND” or “OR”.

2.1.3 Vector Model

The binary assignment of index terms to documents and queries used in the Boolean model is often too limited. Furthermore, the term assignment strategy can be either narrow or broad, i.e. either only strongly related or where (almost) all related terms are assigned. Despite the chosen assignment strategy, the foundation for the evaluation model is limited to the choices made in the assignment process, i.e. the quality of the retrieval process is based on the information available in the descriptions defined by the indexing process. If this uncertainty or graduation of relatedness of terms to the documents is postponed to the evaluation process, this information could be used in the evaluation, and graduate the result not only on behalf of the terms in common, but also on behalf of the term's importance for query and documents. This has been shown to have significant positive influence on the quality of the retrieval [Salton and Lesk, 1997].

In the *vector model* (or *vector space model*) [Salton and McGill, 1986], documents and queries are represented as vectors:

$$\begin{aligned} \vec{d_k} &= (w_{k,1}, w_{k,2}, \ldots, w_{k,t}) \\ \vec{q} &= (w_{q,1}, w_{q,2}, \ldots, w_{q,t}) \end{aligned}$$

where $\vec{d_k}$ represents document d_k and $\vec{q}$ represents query q. The weight $w_{k,i}$ is the term weight for the i'th term of the index vocabulary for document k, where $w_{k,i} \geq 0$ and t is the number of terms in the index vocabulary. The weights in the vector model are therefore not bound to the binary scheme, but can take any positive value, and documents and queries are represented as t-dimensional vectors. For the evaluation process to calculate the degree of similarity between document d_k and query q the correlation between $\vec{d_k}$ and $\vec{q}$ has to be calculated. This is done by a *ranking function* that for a given query (vector) assigns all documents with a weight describing the correlation with the query. Documents can then be ordered by this *weight of correlation* in descending order, thus showing the most relevant documents first. The value of similarity or relatedness expressed by the ranking function is often referred to as the *retrieval status value* (RSV).

All ranking functions of the vector model family are based on the *inner product* [Kraaij, 2004]:

$$RSV(\vec{q}, \vec{d_k}) = \sum_{i=1}^{t} w_{q,i} \cdot w_{k,i} \tag{2.4}$$

and if binary weighting is used defining only presence/absence with term weights as either 1 or 0, then the inner product (2.4) equals the best match evaluation described in the previous section. Binary vectors can be seen as simple crisp sets, while the inner product can be seen as the cardinality of the intersection.

The assignment of weights to documents and queries should reflect the relevance of the given term[3]. A simple and classical example of non-binary term weights is a weight where the number of occurrences of a given term in a document is used as a measure for the relevance of the term to the given document.

In this case, one could argue that the ranking function also has to take into account the size of the documents, which is not the case in the inner

[3] See Section 2.1.1 for more on term weights.

product ranking function (2.4). For instance, take two documents d_1 and d_2, which are one page and 20 pages long, respectively. If both match the same subset of a given query with the same RSV score in (2.4), then it could be argued that the smallest ones are more focused on the query concepts, and thus should be more similar to the query. One approach that takes this into account would be a normalization of the RSV score of (2.4) with the length of the documents defined by the length of the vectors describing them, thus a *vector length normalization*, which corresponds to the cosine of the angle between the vectors [Baeza-Yates and Ribeiro-Neto, 1999] is:

$$RSV(\vec{q}, \vec{d_k}) = cos(\vec{q}, \vec{d_k}) = \frac{\vec{q} \times \vec{d_k}}{||\vec{q}|| \cdot ||\vec{d_k}||} = \frac{\sum_{i=1}^{t} w_{q,i} \cdot w_{k,i}}{\sqrt{\sum_{i=1}^{t} w_{k,i}^2}\sqrt{\sum_{i=1}^{t} w_{q,i}^2}} \tag{2.5}$$

This ranking function is called the *cosine coefficient* and is one of the preferred ranking functions in many vector models[4].

The main advantages of the vector model are the term weighting scheme, the partial matching, and the ranking of documents according to their degree of similarity, which in contrast to the Boolean model, are improvements. The vector model is also simple and has a resilient ranking strategy with general collections [Baeza-Yates and Ribeiro-Neto, 1999]. The main disadvantage is that the conceptual understanding of the ranking is less intuitive, and the ranked answer sets are therefore difficult to improve upon, as it is nearly impossible for users to verify how the ranking is calculated. Naturally, techniques such as relevance feedback and query expansion can be used, but the transparency of the Boolean model is vanished.

2.1.4 Probabilistic Model

The intrinsic uncertainty of information retrieval is based on the problem that it is difficult to distill the meaning from requests and documents, as well as to infer whether a document is relevant to a given request. To deal with the uncertainty, one obvious solution is probability theory, thus estimating the probability that a given document d_k will be relevant with respect to a specific query q, denoted as $P(R|q, d_k)$, where R is a relevance judgment.

The first probabilistic model was introduced in 1960 [Maron and Kuhns, 1997], while the model discussed here was introduced in 1976 [Robertson and Jones, 1976], and was later known as the *binary independence retrieval model.* For the binary independence model, the judgment of a document's relevance to a given query is binary, hence the judgment of documents is either relevant R or not relevant $\bar{R}$.

[4] See [Salton and McGill, 1986] for more on the variety of ranking functions.

The fundamental idea in classical probabilistic retrieval is that term distributions are different for relevant and non-relevant documents, an idea known as the *cluster hypothesis* [Van Rijsbergen, 1979]. Documents and queries are represented as binary vectors in the binary independence model, so instead of estimating $P(R|q, d_k)$ for a specific document d_k and a given query q, the actual estimation is $P(R|\vec{q}, \vec{d_k})$, the probability that one vector of terms (the document) is relevant given another vector of terms (the query). Hence, different documents represented by the same vector of terms yields the same probability of relevance [Fuhr, 1992].

The similarity between a query q and a document d_k is defined as the ratio of the probability of d_k being relevant and the probability of d_k being non-relevant

$$RSV(q, d_k) = \frac{P(R|\vec{q}, \vec{d_k})}{P(\bar{R}|\vec{q}, \vec{d_k})}$$

Let $P(R|\vec{d_k})$ be the probability that $\vec{d_k}$ is relevant to query q, and let $P(\bar{R}|\vec{d_k})$ be the probability that it is non-relevant, and apply Bayes' theorem [Crestani *et al.*, 1998]

$$RSV(q, d_k) = \frac{P(R|\vec{d_k})}{P(\bar{R}|\vec{d_k})} = \frac{P(\vec{d_k}|R) \times P(R)}{P(\vec{d_k}|\bar{R}) \times P(\bar{R})} \tag{2.6}$$

then $P(R)$ and $P(\bar{R})$ would be the prior probability of relevance and non-relevance, and $P(\vec{d_k}|R)$ and $P(\vec{d_k}|\bar{R})$ the probability of observing $\vec{d_k}$, contingent on whether this relevance or non-relevance has been observed.

The goal of the retrieval process is to rank documents in descending order according to the probability of being relevant to a given query, the so-called *probability ranking principle* [Robertson, 1997]. In the probability ranking principle, it is the ranking of the documents that is important, not the actual value of probability, and since $P(R)$ and $P(\bar{R})$ for a given query q are the same for all documents in the collection, equation (2.6) can be reduced to [Crestani *et al.*, 1998]:

$$RSV(q, d_k) \sim \frac{P(\vec{d_k}|R)}{P(\vec{d_k}|\bar{R})} \tag{2.7}$$

The next step is to estimate $P(\vec{d_k}|R)$ and $P(\vec{d_k}|\bar{R})$. In order to simplify this estimation, the components of $\vec{d_k}$ are assumed to be stochastically independent while conditionally dependent upon R and $\bar{R}$. The joint probability distribution of the terms in document d_k is then given by [Crestani *et al.*, 1998]:

$$P(d_k|R) = P(\vec{d_k}|R) = \prod_{i=1}^{t} P(d_{k,i}|R)$$

and

$$P(d_k|\bar{R}) = P(\vec{d_k}|\bar{R}) = \prod_{i=1}^{t} P(d_{k,i}|\bar{R})$$

where t is the number of elements in $\vec{d_k}$. This *binary independent assumption* is the basis for the original binary independence retrieval model [Robertson and Jones, 1976]. This assumption has always been recognized as unrealistic, while the assumption that underpins the binary independence retrieval model was later defined as the weaker *linked assumption* [Cooper, 1991]:

$$\frac{P(\vec{d_k}|R)}{P(\vec{d_k}|\bar{R})} = \prod_{i=1}^{t} \frac{P(d_{k,i}|R)}{P(d_{k,i}|\bar{R})} \tag{2.8}$$

which states that the ratio between the probabilities of $\vec{d_k}$ occurring in relevant and non-relevant documents is equal to the product of the corresponding ratios of the single elements.

The product of equation (2.8) can be split according to the occurrences of terms in document d_k [Fuhr, 1992]:

$$\frac{P(\vec{d_k}|R)}{P(\vec{d_k}|\bar{R})} = \prod_{i=1}^{t} \frac{P(d_{k,i} = 1|R)}{P(d_{k,i} = 1|\bar{R})} \prod_{i=1}^{t} \frac{P(d_{k,i} = 0|R)}{P(d_{k,i} = 0|\bar{R})} \tag{2.9}$$

Recalling that $P(d_{k,i}|R) + P(d_{k,i}|\bar{R}) = 1$ and using a logarithmic transformation to obtain a linear RSV function, the expression, in the end, for ranking computation in the probabilistic model, while ignoring factors which are constant for all documents in the context of the same query, would be [Baeza-Yates and Ribeiro-Neto, 1999]:

$$RSV(q, d_k) \sim \sum_{i=1}^{t} w_{q,i} \times w_{k,i} \times \left(\log \frac{P(d_{k,i}|R)}{1 - P(d_{k,i}|R)} + \log \frac{P(d_{k,i}|\bar{R})}{1 - P(d_{k,i}|\bar{R})} \right)$$

An estimation of $P(d_{k,i}|R)$ and $P(d_{k,i}|\bar{R})$ is necessary to apply the binary independence model. Before any documents are retrieved a simplifying assumption must be made, such as:

$$\begin{array}{rcl} P(d_{k,i}|R) & = & 0.5 \\ P(d_{k,i}|\bar{R}) & = & \frac{n_i}{N} \end{array}$$

where $P(d_{k,i}|R)$ is a constant for all index terms, typically 0.5, and n_i is the number of documents which contain the term $d_{k,i}$ and N the total number of documents in the collection.

Let V be a subset of the documents retrieved and ranked by the probabilistic model, defined, for instance, as the top r ranked documents, where r is a previously defined threshold. Let V_i be the subset of V which contains the term $d_{k,i}$. Then the estimate of $P(d_{k,i}|R)$ and $P(d_{k,i}|\bar{R})$ can be calculated as

$$\begin{array}{rcl} P(d_{k,i}|R) & = & \frac{|V_i|}{|V|} \\ P(d_{k,i}|\bar{R}) & = & \frac{n_i-|V_i|}{N-|V|} \end{array}$$

where $|X|$ is the cardinality of the set X. This can then be repeated recursively, and the initial guess of the probabilities of $P(d_{k,i}|R)$ and $P(d_{k,i}|\bar{R})$ can then be improved without any assistance from users. However, relevance feedback could also be used after the first guess [Baeza-Yates and Ribeiro-Neto, 1999].

On a conceptual level the advantage of the probabilistic model is that documents are ranked in descending order according to their probability of being relevant, based on the assumption that this is presumably easier for most users to understand, rather than, for instance, a ranking by the cosine coefficient [Cooper, 1991]. The probabilistic model has not been widely used; where it has been used, however, it has achieved retrieval performance (in the sense of quality) comparable to, but not clearly superior to, non-probabilistic methods [Cooper, 1991]. The major disadvantages of the binary independence retrieval model are that documents are either to be considered relevant or not, and that the term weighting is binary. The relevance judgment depends on the assumptions made about the distribution of terms, e.g., how terms are distributed throughout documents in the set of relevant documents, and in the set of non-relevant documents, which is usually defined statistically. As mentioned earlier, term weighting has been shown to be very important for the retrieval of information, which is not supported by the binary independence retrieval model. Last, the model must somehow estimate the initial probabilities of $P(d_{k,i}|R)$ and $P(d_{k,i}|\bar{R})$, either by (qualified) guessing or by a manual estimation done by experts. Later research has shown probability models that overcome some of the major disadvantages, e.g. [Van Rijsbergen, 1986; Robertson and Walker, 1997; Ng, 1999].

2.1.5 Fuzzy Set Model

The Boolean model is the only one of the three classical models presented above that has the option of controlling the interpretation of queries beyond the weighting of keywords. In the vector and probabilistic models, queries are assumed to be weighted vectors of keyword, whereas queries in the Boolean model are Boolean expressions with a well-defined logic. Naturally, as mentioned above, users have difficulties using Boolean expressions, a problem vector and probabilistic models solve by not offering any operators for the query language. Query languages without operators are preferable due to their simplicity, but are at the same time limited, and as users become more experienced, they may benefit from more expressive query languages.

One natural alternative solution to the Boolean model is the *fuzzy set model* which generalizes the Boolean model by using *fuzzy sets* [Zadeh, 1993] instead of *crisp sets* (*classical sets*). The fuzzy set model can handle queries of Boolean expressions [Bookstein, 1980] as well as queries of weighted vectors of keywords without operators, like the vector and probabilistic models [Yager, 1987]. Moreover, the fuzzy set model can deal with the uncertainty of information retrieval, both with regard to term weighting in the representation of documents and the relevance between queries and documents in the evaluation process.

Information searching and retrieval has to deal with information characterized, to some extent, by a kind of *imperfection*, where imperfection is defined as imprecision, vagueness, uncertainty, and inconsistency [Motro, 1990]. Imprecision and vagueness are related to the information content of a proposition, uncertainty is related to the truth of a proposition, and inconsistency comes from the simultaneous presence of contradictory information [Bordogna and Pasi, 2001].

Queries can be seen as imperfect expressions of a user's information needs and the representation of documents can be seen as a partial and imprecise characterization of the documents' content. A document should be considered relevant when the document meets the actual user's information needs, and can therefore be seen as a matter of *degree of relevance*, a possibility of being relevant, rather than a probability of being relevant. Hence, there is only an uncertain relationship between the request and the likelihood that the user would be satisfied with that response [Lucarella, 1990]. The *fuzzy set theory* provides a sound mathematical framework for dealing with the imperfection of documents and queries. The substantial difference between models based on probability and possibility, is that the probabilistic view presupposes that a document is or is not relevant for a given query, where the possibilistic view presupposes that the document may be more or less relevant for a given query [Lucarella, 1990].

Basic Operations on Fuzzy Sets

A crisp set is defined by a function, usually called the *characteristic function*, which declares which elements of a given universal set U are members of the set and which are not. Set A is defined by the characteristic function χ_A as follows:

$$\chi_A(u) = \begin{cases} 1 & \text{for } u \in A \\ 0 & \text{for } u \notin A \end{cases} \tag{2.10}$$

and maps all elements of U to elements of the set {0,1}:

$$\chi_A : U \to \{0, 1\}$$

discriminating between members and non-members of a crisp set. A fuzzy set is defined by a *membership function*, a generalized characteristic function, which for a fuzzy set A maps elements from a given universal set U to the unit interval [0,1]:

$$\mu_A : U \to [0, 1]$$

where μ_A defines the membership grade of elements in A. $\mu_A(u) = 1$ denotes full membership, $\mu_A(u) = 0$ denotes no membership at all, and $0 < \mu_A(u) < 1$ denotes partial membership [Klir and Yuan, 1995]. Obviously, the characteristic function in (2.10) is a special case of the fuzzy set membership function, where fuzzy sets are identical to crisp sets.

Finite fuzzy sets are often specified as:

$$A = \{\mu_A(u)/u | u \in U\}$$

or

$$A = \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + \cdots + \mu_A(u_n)/u_n = \sum_{i=1}^{n} \mu_A(u_i)/u_i$$

where + and $\sum$ denotes union.

The *cardinality* of a fuzzy set defined over a finite universe U, called the *scalar cardinality* or the *sigma count*, is the summation of the membership degrees of all $u \in U$ defined as:

$$|A| = \sum_{u \in U} \mu_A(u)$$

for set A [Klir and Yuan, 1995].

An important concept of fuzzy sets is the *α-cut*, which for a given fuzzy set A, defined on the universal set U and with $\alpha \in [0, 1]$, is the crisp set

$$^{\alpha}A = \{x | \mu_A(x) \leq \alpha\}$$

that contains all elements of the universal set U, whose membership grades in A are greater than or equal to the specified value of α.

The *complement*, $\bar{A}$, of a fuzzy set A with respect to the universal set U is defined for all $u \in U$ as:

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

Given two fuzzy sets, A and B, defined in the universe U, the *intersection*, $A \cap B$, and the *union*, $A \cup B$, can be generalized by the use of *triangular-norms* (T-norms) and *triangular-conorms* (T-conorms), respectively:

$$\begin{aligned} \mu_{A \cap B} &= T\left(\mu_A(u), \mu_B(u)\right) \\ \mu_{A \cup B} &= S\left(\mu_A(u), \mu_B(u)\right) \end{aligned}$$

where $\mu_{A \cap B} = T(\mu_A(u), \mu_B(u)) = \min(\mu_A, \mu_B)$ is the standard T-norm function and $\mu_{A \cup B} = S(\mu_A(u), \mu_B(u)) = \max(\mu_A, \mu_B)$ the standard T-conorm function. In Table 2.1, some alternative T-norms and T-conorms are shown [Chen and Chen, 2005]. Whenever we only refer to intersection and union on fuzzy sets without explicating a particular T-norm or T-conorm, the standard versions are assumed.

All the normal operations on crisp sets also apply to fuzzy sets, except for the law of excluded middle, $A \cap \bar{A} = \emptyset$, which for the standard intersection is only true in the special case where fuzzy sets are identical to crisp sets, i.e. when the membership function equals the characteristic function [Cross, 1994].

For any pair of fuzzy sets, defined in a finite universe U, the *degree of subsethood* of A and B is defined as:

$$subsethood(A, B) = \frac{1}{|A|}\left(|A| - \sum_{u \in U} \max\left(0, \mu_A(u) - \mu_B(u)\right)\right) \tag{2.11}$$

where the $\sum$ term in this formula describes the sum of the degree to which the subset inequality $\mu_A(u) \leq \mu_B(u)$ is violated, with the $|A|$ as a normalizing factor to obtain the range $0 \leq subsethood(A, B) \leq 1$. This can be expressed more conveniently in terms of sets:

$$subsethood(A, B) = \frac{|A \cap B|}{|A|} \tag{2.12}$$

as the ratio between the cardinality of the intersection $A \cap B$ and the cardinality of A [Klir and Yuan, 1995].

Fuzzy Information Retrieval

The weighted vector representation of documents in the vector model is easily transformed into fuzzy sets, and so is any other type of term-weighting

T-norms		T-conorms	
$\min(\mu_A, \mu_B)$	Logical Product	$\max(\mu_A, \mu_B)$	Logical Sum
$\mu_A \times \mu_B$	Algebraic Product	$\mu_A + \mu_B - (\mu_A \times \mu_B)$	Algebraic Sum
$\dfrac{\mu_A \times \mu_B}{\mu_A + \mu_B - (\mu_A \times \mu_B)}$	Hamacher Product	$\dfrac{\mu_A + \mu_B - 2(\mu_A \times \mu_B)}{1 - (\mu_A \times \mu_B)}$	Hamacher Sum
$\begin{cases} \mu_A, & \text{if } \mu_B = 1, \\ \mu_B, & \text{if } \mu_A = 1, \\ 0 & \text{otherwise} \end{cases}$	Drastic Product	$\begin{cases} \mu_A, & \text{if } \mu_B = 0, \\ \mu_B, & \text{if } \mu_A = 0, \\ 1 & \text{otherwise} \end{cases}$	Drastic Sum
$\max(\mu_A + \mu_B - 1, 0)$	Bounded Product	$\min(\mu_A + \mu_B, 1)$	Bounded Sum

Table 2.1: *Some T-norms and T-conorms*

schemes. The relation between terms and documents can be expressed as a binary fuzzy indexing:

$$I = \{\mu_I(d,t)/(d,t) | d \in D; t \in T\} \tag{2.13}$$

where D and T are finite sets of documents and terms, respectively, and $\mu_I : D \times T \rightarrow [0,1]$, a membership function indicating for each pair (d,t) the strength of the relation between the term t and document d [Lucarella, 1990].

On the basis of the indexing relation I, it is possible to define descriptions for each document $d \in D$ and each query $q \in Q$:

$$I_d = \{\mu_{I_d}(t)/(t) | t \in T; \mu_{I_d}(t) = \mu_I(d,t)\}$$
$$I_q = \{\mu_{I_q}(t)/(t) | t \in T; \mu_{I_q}(t) = \mu_I(q,t)\}$$

as fuzzy sets. If queries are defined as Boolean expressions, then some considerations have to be made on how to interpret the combination of the (possible) weighting of terms and Boolean operators, as in Bookstein [1980].

The *retrieval rule* can likewise be expressed in the form of a binary fuzzy relation:

$$R = \{\mu_R(q,d)/(q,d) | q \in Q; d \in D\} \tag{2.14}$$

with the membership function $\mu_R : Q \times D \rightarrow [0,1]$. For a given query $q \in Q$, on the basis of the retrieval relation R, the retrieved fuzzy subset R_q of document set D is defined as:

$$R_q = \{\mu_{R_q}(d)/d | d \in D; \mu_{R_q} = \mu_R(q,d)\} \tag{2.15}$$

where $\mu_{R_q}(d)$ represents the strength of the relationship between the document d and the query q, hence a reordering of the collection with respect to the values of $\mu_{R_q}(d)$. The retrieved fuzzy set R_q can be restricted to a given threshold, $\alpha \in [0,1]$, by the α-level cut of R_q:

$$R_{q(\alpha)} = \{\mu_{R_q}(d)/d | \mu_{R_q}(d) \geq \alpha; d \in D; \mu_{R_q} = \mu_R(q,d)\} \tag{2.16}$$

which define that the extracted elements should have a membership value greater than or equal to the fixed threshold α. The elements in $R_{q(\alpha)}$ would then be those documents that best fulfill the query q to the degree defined by α. A ranked output can be returned to the user by arranging the retrieved documents in descending order according to the degree of the retrieval status value $RSV = \mu_{R_q}(d)$.

The computation of the RSV for documents given a specific query has a variety of different solutions. A simple approach is to interpret the queries with a conjunctive or a disjunctive reading of the terms. In Cross [1994] two different approaches based on set-theoretic inclusion are given:

$$RSV = \mu_{R_q}(d) = \mu_R(q,d) = \frac{\sum_{t \in T} \min\left(\mu_{I_q}(t), \mu_{I_d}(t)\right)}{\sum_{t \in T} \mu_{I_q}(t)} \tag{2.17}$$

and

$$RSV = \mu_{R_q}(d) = \mu_R(q,d) = \frac{\sum_{t \in T} \min\left(\mu_{I_q}(t), \mu_{I_d}(t)\right)}{\sum_{t \in T} \mu_{I_d}(t)} \tag{2.18}$$

defining the $subsethood(q,d)$ of query q in document d and the $subsethood(d,q)$ of document d in query q, respectively. In (2.17) and (2.18), the length of the query and the documents, respectively, are used as normalization to obtain that $0 \leq RSV \leq 1$, but could also be argued for based upon the same reasons as for the normalization introduced by the cosine coefficient (2.5) in the vector model.

A different kind of RSV approach is by means of an *averaging operator*, which provides an aggregation between *min* and *max*, where the simple (arithmetic) mean is defined as :

$$RSV(q,d) = \mu_{R_q}(d) = \mu_R(q,d) = \frac{1}{|I_q|} \sum_{t \in T} \frac{\mu_{I_q}(t) + \mu_{I_d}(t)}{2} \tag{2.19}$$

the harmonic mean as

$$RSV(q,d) = \mu_{R_q}(d) = \mu_R(q,d) = \sum_{t \in T} \frac{2\mu_{I_q}(t)\mu_{I_d}(t)}{\mu_{I_q}(t) + \mu_{I_d}(t)} \tag{2.20}$$

and the geometric mean as

$$RSV(q,d) = \mu_{R_q}(d) = \mu_R(q,d) = \sum_{t \in T} \sqrt{\mu_{I_q}(t)\mu_{I_d}(t)} \tag{2.21}$$

for query q and document d. All these means provide different fixed views on the average aggregation. A natural generalization would be to introduce parameterized averaging operators that can scale the aggregation between *min* and *max*.

Ordered Weighted Averaging Operator

An important parameterized averaging operator is the *Ordered Weighted Averaging Operator* (OWA) [Yager, 1988], which takes a weighted "ordering" vector as a parameter, instead of a single value. For examples of different parameterized averaging approaches see e.g. [Waller and Kraft, 1979; Salton *et al.*, 1983; Smith, 1990].

An ordered weighted averaging operator of dimension n is a mapping $OWA : [0,1]^n \rightarrow [0,1]$, which has an associated weighting vector $W = (w_1, w_2, \ldots, w_n)$ such that:

$$\sum_{j=1}^{n} w_j = 1$$

and each $w_j \in [0,1]$. The aggregation with respect to a weighting vector, W, is defined as:

$$OWA_W(a_1, a_2, \ldots, a_n) = \sum_{j=1}^{n} w_j b_i$$

with b_i being the j'th largest of the a_i. Hence, the weights in W are associated to a particular position rather than a particular element.

The generality of the ordered, weighted averaging operator lies in the fact that the structure of W controls the aggregation operator, and the selection of weights in W can emphasize different arguments based upon their position in the ordering. If most of the weights are put in the beginning of the weighting vector, emphasis is put on the higher scores, while placing them near the end of W emphasizes the lower scores in the aggregation. The upper and lower bounds of the ordered weighted averaging operator are defined by the weighting vectors W^*, where $w_1 = 1$ and $w_j = 0$ for $j \neq 1$, and W_*, where $w_n = 1$ and $w_j = 0$ for $j \neq n$, for n dimensions and correspond to *max* and *min*, respectively. Another interesting weighting vector is W_{ave}, where $w_j = \frac{1}{n}$ for all n and which corresponds to the simple mean (2.19) [Yager, 2000]. Obviously, the variety of weighting vectors between the lower and upper bounds are infinite, while there is only one for each of the bounds. A measure for characterizing OWA operators, called the *alpha value* of the weighting vector, is defined as:

$$\alpha = \frac{1}{n-1}\sum_{j=1}^{n}(n-j)w_j$$

where n is the dimension and w_j the j'th weight in the weighting vector. It can be shown that $\alpha \in [0,1]$, and furthermore that $\alpha = 1$ if $W = W^*$, $\alpha = 0.5$ if $W = W_{ave}$, and $\alpha = 0$ if $W = W_*$ [Yager, 2000].

The weighting vector can be obtained by a regularly increasing monotonic function which conforms to $f(0) = 0$, $f(1) = 1$, and if $r_1 > r_2$ then $f(r_1) \geq f(r_2)$. An example is given here of such a function, which takes α as input:

$$W_j = \left(\frac{j}{n}\right)^{\frac{1}{\alpha}-1} - \left(\frac{j-1}{n}\right)^{\frac{1}{\alpha}-1} \tag{2.22}$$

and distributes the weights in the weighting vector according the description above, with more emphasis in the beginning of W when $\alpha \rightarrow 1$, simple mean when $\alpha = 0.5$, and more emphasis in the end of W, when $\alpha \rightarrow 0$, as shown in Table 2.2.

α	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	0.0
W	1.00	0.88	0.76	0.62	0.48	0.33	0.19	0.08	0.02	0.01	0.00
	0.00	0.07	0.14	0.22	0.28	0.33	0.35	0.31	0.18	0.02	0.00
	0.00	0.05	0.10	0.16	0.24	0.33	0.46	0.61	0.80	0.97	1.00

Table 2.2: *An example of how to distribute the weights by a function*

The function in (2.22) is an example of how to parameterize n-dimensional weighting vectors with a single value. The resulting vector will always correspond to an averaging operator. Some weighting vectors can meaningfully be associated with linguistic quantifiers, e.g. W^* correspond to *there exists* and W_* to *for all*. Linguistic quantifiers could also be defined as regularly increasing monotonic functions

$$W_j = \mathcal{Q}\left(\frac{j}{n}\right) - \mathcal{Q}\left(\frac{j-1}{n}\right) \tag{2.23}$$

where $\mathcal{Q}$ is some function defining some quantifier. This would permit users to define weighting vectors from a selection of well-known linguistic qualifiers, e.g. *all*, *most*, *some*, *few*, *at least one*, corresponding to the proportion of elements in the queries deemed important. If a query evaluation with the quantifier *few* gives an enormous result set, the same query with the quantifier *most* could be used to restrict the query evaluation, and vise versa.

2.2 Retrieval Evaluation

The process of querying a retrieval system can be evaluated as a complete task, starting with posing the query and continuing until satisfactory information has been found, including any intermediate tasks, for instance, the interactive process of possible reformulations of the initial request.

These kinds of evaluations are sometimes called *real-life evaluation*, in contrast to evaluation that only compares the query and the result, which often is referred to as *laboratory evaluation.* Over the last couple of decades, there has been more focus on real-life evaluation (see e.g. [Ingwersen, 1992]), but this kind of evaluation is more complicated and expensive, thus laboratory evaluation is still most prevalent. Laboratory evaluation is also the kind of retrieval evaluation used in this thesis.

The efficiency of an information retrieval process is commonly measured in terms of *recall* and *precision.* Recall is the ratio between the number of relevant documents retrieved to the total number of relevant documents, and precision is the ratio between the number of relevant documents retrieved to the total number of retrieved documents. Let R be the set of relevant documents associated to a given query and A the retrieved set of documents from posing the query, meaning recall and precision can be defined as:

$$\begin{aligned} recall &= \tfrac{|A \cap R|}{|R|} \\ precision &= \tfrac{|A \cap R|}{|A|} \end{aligned}$$

where $|x|$ denotes the cardinality of the set x. Recall and precision, as defined above, assume that all documents in the retrieved set A have been seen. However, the user is not usually presented with all documents in A, but instead A is sorted according to the degree of relevance (or the probability of being relevant), and the user thus only sees a fraction of A.

For example, consider the relevant set of a document to a given query q, defined a priori by some experts, as the following set R:

$$R = \{d_3, d_7, d_{12}, d_{24}, d_{29}, d_{71}, d_{83}, d_{115}, d_{141}, d_{195}\}$$

and the following result A as the result of posing query q to a given retrieval system

1. d_3 •
2. d_37
3. d_{78} •
4. d_{12}
5. d_6

6. d_{24} •
7. d_{117}
8, d_{49}
9. d_{141} •
10. d_{27}

11. d_9
12. d_{68}
13. d_{155}
14. d_{219}
15. d_{195} •

where the relevant documents in A are marked with a bullet. Instead of considering the complete result set A, recall and precision can be computed for any fragment of A. Let A_i be the fragment of A with the i topmost answers. Recall and precision on, for instance, A_1 would then be $1/10 = 10\%$ and $1/1 = 100\%$, respectively. Normally, a set of *standard recall levels*, which are $0\%, 10\%, 20\%, \ldots, 100\%$, are used to plot a recall and precision curve. Here is the computation for the above result:

$$\begin{array}{lll} & \textit{Recall} & \textit{Precision} \\ A_1 & 1/10 = 10\% & 1/1 = 100\% \\ A_3 & 2/10 = 20\% & 2/3 \approx 66\% \\ A_6 & 3/10 = 30\% & 3/6 = 50\% \\ A_9 & 4/10 = 40\% & 4/9 \approx 44\% \\ A_{15} & 5/10 = 50\% & 5/15 \approx 33\% \end{array} \tag{2.24}$$

for recall from $10 - 50\%$. The plot for this computation is shown in Figure 2.5. In this example, the recall and precision are for a single query. Usually, however, retrieval algorithms are evaluated by running them for several distinct queries, and an average is used for the recall and precision figures.

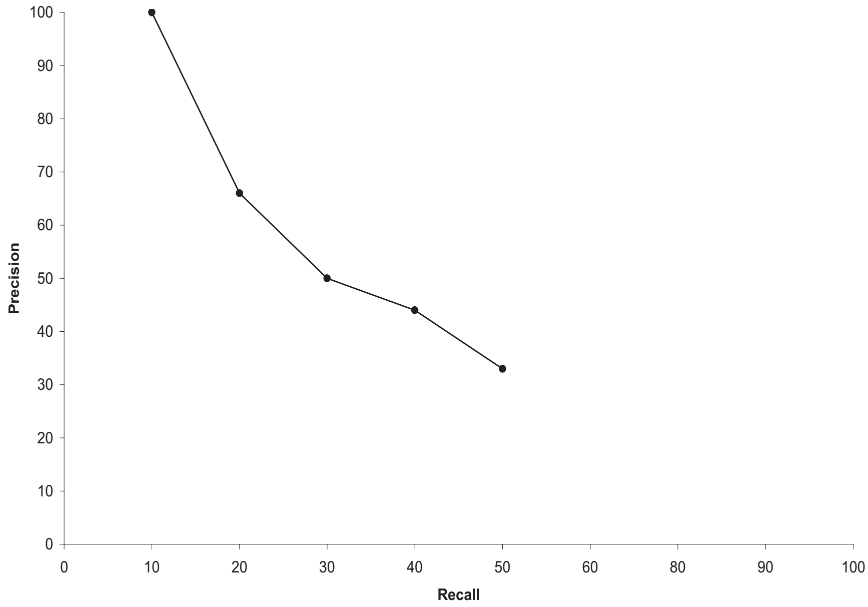


Figure 2.5: *Recall and precision for the example query q in* (2.24)

This way, recall and precision can be used on ranked retrieval, but only the position in A is used in the evaluation. In case of *vague ordering*, where the degree of relevance is not only bound to the position, but also to the actual degree associated to the documents, then recall and precision might be inadequate [Baeza-Yates and Ribeiro-Neto, 1999].

The output of a fuzzy retrieval system is a list of all the documents ranked according to their relevance evaluations. The retrieval status value is the membership grade of the retrieved fuzzy set of relevant documents. A fuzzy concept of retrieval is needed that catches the vagueness captured in the result.

One obvious solution to this is to generalize recall and precision to fuzzy sets instead of crisp sets. Recall and precision can then be redefined as:

$$\begin{aligned} recall_{fuzzy} &= \tfrac{|A_f \cap R_f|}{|R_f|} \\ precision_{fuzzy} &= \tfrac{|A_f \cap R_f|}{|A_f|} \end{aligned}$$

where R_f is the relevant documents for a given query as a fuzzy set, and A_f the vague ordering retrieval. R_f is supposed to be defined a priori by experts, like R in normal recall and precision. This can also be described as $recall = subsethood(R, A)$ and $precision = subsethood(A, R)$ [Buell and Kraft, 1981].

Another measure sometimes also used in the evaluation is the *fallout*, the fraction of non-relevant documents that are retrieved:

$$fallout = \frac{|A \cap \bar{R}|}{|\bar{R}|}$$

where $\bar{X}$ is the complement. The fallout measure can naturally also be generalized to fuzzy sets, since fuzzy sets support the complement operation.

2.3 Summary and Discussion

In this chapter, three classical information retrieval models have been presented, one based on Boolean logic and two alternative models based respectively on vector space and probability. Characteristic of operational information retrieval systems until the mid-nineties was that they, almost without exception, adopted the Boolean model of searching. However, in the last couple of decades, the vector retrieval model has gained some ground, while the probabilistic retrieval model is still rarely in use.

The three classical models share a rather straightforward representation paradigm, “bag of words”, for representing the content of documents and queries (except for the Boolean model, where queries are Boolean expressions – a disjunction of conjunctive vectors). Each index term in these vectors can have a weight attached denoting the relevance of the term describing the content of a document or query.

The aim in this thesis is to introduce external knowledge and in so doing promote semantics in the retrieval process. This is to be achieved by introducing concept models in the form of ontologies into information retrieval.

Naturally, this requires changes in both representation and query evaluation. The recognition of semantics in documents and queries involves some kind of natural language processing, which as result has a more complex representation paradigm. The classical framework seems unfeasible for adapting to this kind of representation, likewise for the classical query evaluation with search strategies based solely on the occurrence of string sequences, terms or words, or combinations of words. The fuzzy set retrieval model is thus deemed a better alternative.

The fuzzy set retrieval model is a generalization of the Boolean model, and thus has a well-defined underlying logical skeleton. This framework is flexible enough to capture the paradigm shift of representations from simple collections of words to semantic descriptions. Furthermore, it underpins the notion of imprecision perfectly[5].

The fuzzy set model is only rarely used, but appears to be an appropriate way towards solving the problems concerning ontology-based information retrieval dealt with in this thesis. The flexibility of the aggregation, especially hierarchical aggregation, and the use of linguistic quantifiers as parameterization of the aggregation process, are shown to substantiate the selection of the fuzzy set retrieval model in Section 6.2.3.

[5] This is also true for the vector and probabilistic models, while the Boolean model hands the question of uncertainty over to users.

Chapter 3

Ontology

Up until the last decade of the 20^{th} century, ontology had primarily been a discipline in philosophy dealing with the nature and organization of reality. In recent years, however, ontology has also emerged as a research area related to computer science. Two main, corresponding definitions of the word ”ontology” can be found in various dictionaries, for instance, Webster's defines it as[Webster, 1993]:

1. *A science or study of being: specifically, a branch of metaphysics relating to the nature and relations of being; a particular system according to which problems of the nature of being are investigated; first philosophy.*

2. *A theory concerning the kinds of entities and specifically the kinds of abstract entities that are to be admitted to a language system.*

where the first meaning applies to the philosophical discipline, and the second to the branch of computer science known as knowledge engineering. This twofold definition of ontology is concurrent with the different typing suggested by Gurino and Giaretta [1995], where “Ontology” and “ontology” refer to the philosophy and knowledge engineering definitions, respectively.

While the above distinction is adopted in this thesis, only the second definition, which refers to the knowledge engineering sense of the word, is used beyond this section.

3.1 Ontology as a Philosophical Discipline

In philosophy, ontology is an important area whose origins date back more than 2000 years, the plethora of discussions and opinions that have been put forward over the centuries making it difficult to present a brief historical overview that is not superficial. Hence, only a select number of milestones related to this thesis will be pointed out.

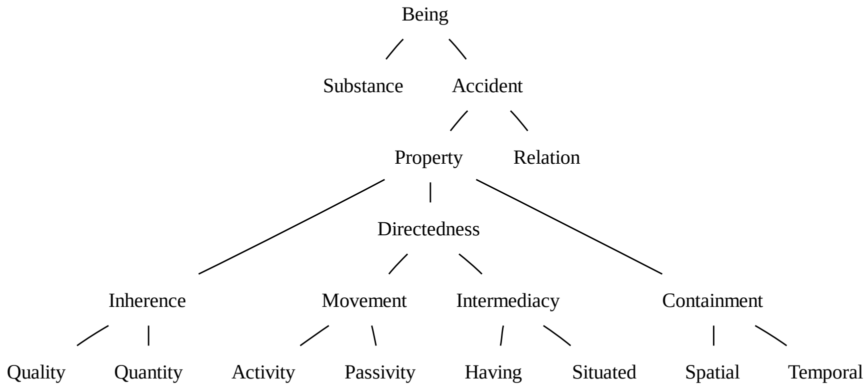


Figure 3.1: *Brentano's tree of Aristotle's categories*

Aristotle defined ontology as the science of being. In *Categories*, Aristotle presented ten basic categories for classifying things: *Substance*, *Quality*, *Quantity*, *Relation*, *Activity*, *Passivity*, *Having*, *Situatedness*, *Spatiality*, and *Temporality.* Later, the philosopher Franz Brentano organized all ten categories as leaves on a single tree, as shown in Figure 3.1, where the branches are labeled with terms taken from some of Aristotle's other works. [Sowa, 2000].

Quantity	**Quality**	**Relation**	**Modality**
Unity	Reality	Inherence	Possibility
Plurality	Negation	Causality	Existence
Totality	Limitation	Community	Necessity

Table 3.1: *Kant's categories*

This classification was widely accepted until Emmanuel Kant (1724-1804) challenged Aristotle's idea that the essence of things is solely determined by the things themselves. Kant's idea was that the question of essence could not be separated from whoever perceives and understands the thing concerned. Kant, for whom a key question was what structures the mind uses to capture reality, organized his categories in four classes, each of which presents a triadic pattern, as shown in Table 3.1. Kant obtained his categories from the logic classification of judgments. Thus, for instance, *unity* matches singular judgments, *plurality* matches particular judgments, and *totality* matches universal judgments. The sensations involved when a person perceives reality are put into order, first in space and time, and then, according to categories. [Gómez-Pérez *et al.*, 2004]

Charles S. Peirce (1839-1914) found that some of Kant's triads reflected three additional basic categories: *Firstness*, *Secondness*, and *Thirdness*. *Firstness* is determined by a thing's inherent qualities, *Secondness* is determined in relation to something else, and *Thirdness* by mediation that brings multiple entities into relationship. The first can be defined by a monadic predicate P(x), the second by a dyadic relation R(x, y), and the third by a triadic relation M(x, y, z). For example, the type *human* can be defined by the quality inherent in the individual, the type *mother* is defined in relation to a *child*, and the type *motherhood* relates *mother* and *child* [Sowa, 2000].

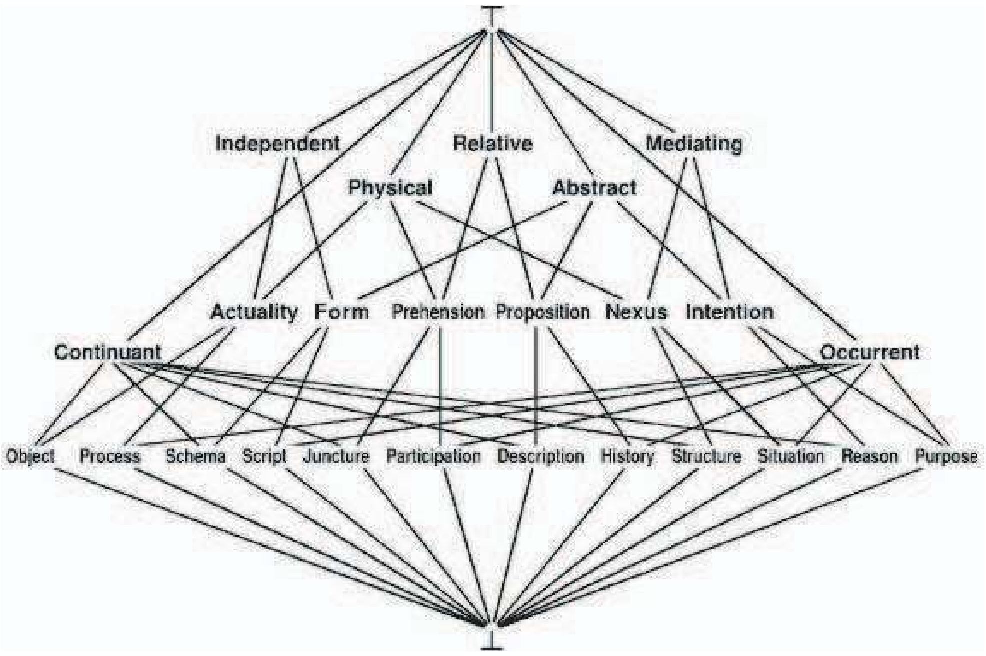


Figure 3.2: *Sowa's lattice of categories*

Alfred North Whitehead (1861-1947) developed an ontology that combined the insights of some of history's greatest philosophers. Even though he never mentioned Peirce, there was a great deal of overlap with some of Peirce's triads in some of Whitehead's eight categories [Sowa, 2000]. The first tree *actual entries*, *prehensions*, and *nexūs* correspond to Peirce's *Fristness*, *Secondness*, and *Thirdness*. In addition to these three physical categories, he defined three categories for abstractions, *eternal objects*, *propositions*, and *subjective forms*, constituting a triad for abstractions. Whitehead's last two categories are principles for the construction of new categories, *multiplicities* and *contrasts* [Sowa, 2000].

The aspects of ontology described up to this point could very well be defined as general ontology, in contrast to various special or domain-specific ontologies, which are also defined as formal and material ontologies by Husserl [Guarino and Giaretta, 1995]. In his book, *Knowledge Representation*, John F. Sowa merges most of the above-mentioned ideas into a lattice of categories, shown in Figure 3.2, which are an example of a top ontology [Sowa, 2000].

3.2 Knowledge Engineering Ontologies

After this short introduction to the philosophical definition of ontologies, we now turn to its primary meaning, which is the knowledge engineering definition. In their paper "Ontology and Knowledge Bases" Gurino and Giaretta [1995], list the following seven different interpretations of the term "ontology":

1. Ontology as a philosophical discipline,
2. Ontology as a an informal conceptual system,
3. Ontology as a formal semantic account,
4. Ontology as a specification of a "conceptualization",
5. Ontology as a representation of a conceptual system via a logical theory,
   - 5. 1 characterized by specific formal properties,
   - 5. 2 characterized only by its specific purposes,
6. Ontology as the vocabulary used by a logical theory,
7. Ontology as a (meta-level) specification of a logical theory.

The first interpretation of ontology is as described in the previous section and differs from the six others, which are all related to the knowledge engineering sense of ontology. Gurino and Giaretta divide the remaining six interpretations (2-7) into two subdivided groups:

1. A particular framework at the semantic level (interpretations 2-3), and
2. An ontology intended as a concrete artifact at the syntactic level (interpretations 4-7).

They support the ambiguity between these two interpretations and explain that one technical term cannot cover both of them. Their suggestion is to define the first group as *conceptualization* and the second as *ontological theory*, denoting a semantic structure which reflects a particular conceptual system, as well as a logical theory intended to express ontological knowledge, respectively.

The notion of the second group, *ontological theory*, is in fact designed artifacts, the interpretation for which the term "ontology" is primarily used in this thesis. This interpretation is compatible with Gruber's definition of ontology (with a lowercase "o"), the most quoted in the literature:

> *An ontology is an explicit specification of a conceptualization.*

[Gruber, 1993][1]. Actually, Gruber's definition is one of the interpretations put forward by Gurino and Giaretta in their list of different interpretations.

Gómez-Pérez et al. [2004] conclude that ontologies aim to capture consensual knowledge in a generic way. They also state that despite the many different interpretations of the term "ontology", there is consensus within the community, so confusion is avoided.

3.3 Types of Ontologies

Ontologies can be classified from at least two aspects, by the type of information they capture, and by the richness of their internal structure. The former is divided into six groups in [Gómez-Pérez *et al.*, 2004]:

Top-level ontologies or upper-level ontologies are the most general ontologies describing the top-most level in ontologies to which all other ontologies can be connected, directly or indirectly. In theory, these ontologies are shareable as they express very basic knowledge, but this is not the case in practice, because it would require agreement on the conceptualization of being, which, as described in Section 3.1, is very difficult. Any first step from the top-most concept would be either too narrow, thereby excluding something, or too broad, thereby postponing most of the problem to the next level, which means of course that controversial decisions are required.

Domain ontologies describe a given domain, e.g. medicine, agriculture, politics, etc. They are normally attached to top-level ontologies, if needed, and thus do not include common knowledge. Different domains can be overlapping, but they are generally only reusable in a given domain. The overlap in different domains can sometimes be handled by a so-called middle-layer ontology, which is used to tie one or more domain ontologies to the top-level ontology.

Task ontologies define the top-level ontologies for generic tasks and activities.

Domain-task ontologies define domain-level ontologies on domain-specific tasks and activities.

[1] Gurino and Giaretta discuss the problem with the external definition of conceptualization in this definition in their paper [Guarino and Giaretta, 1995], which is not looked at here.

Method ontologies give definitions of the relevant concepts and relations applied to specify a reasoning process so as to achieve a particular task.

Application ontologies define knowledge on the application-level and are primarily designed to fulfill the need for knowledge in a specific application.

This is similar to the definition given in [Guarino, 1998], shown in Figure 3.3. The only difference is that this model is less fine-tuned. Even though the different ontologies are connected in this figure, they can be used as individual ontologies, or can be connected in any number of combinations. However, the ordering is normally respected, e.g. an "application ontology" is hardly ever connected directly to a "top-level ontology".

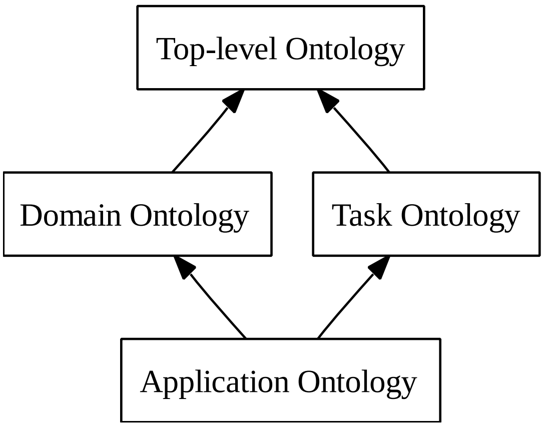
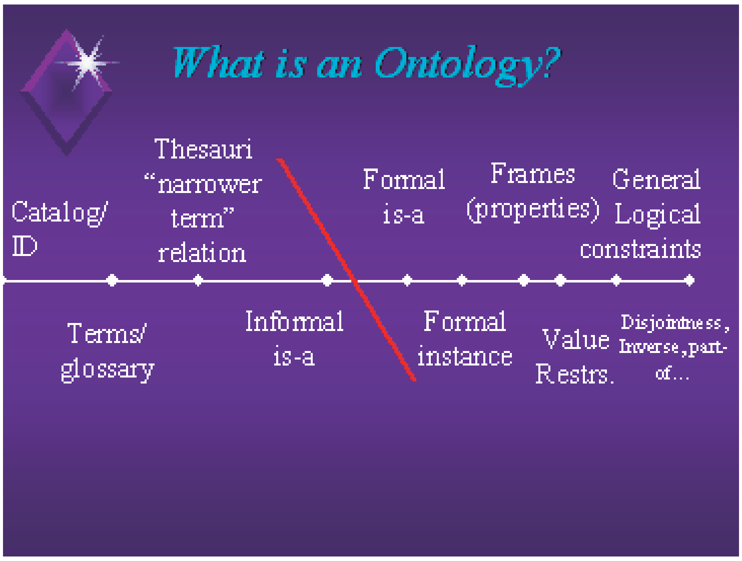


Figure 3.3: *Guarino's kinds of ontologies*

Another type of ontology which should be mentioned here is the so-called linguistic ontologies, whose origin is natural languages and which describe semantic constructs rather than model a specific domain. Most of the linguistic ontologies use words as a grammatical unit, and can be used for natural language processing and generation. The mapping between units in natural language and the meanings in the linguistic ontologies are obviously not one-to-one due to fact that the meanings of grammatical units in natural languages can be ambiguous. Linguistic ontologies are quite heterogeneous and are often combined with top-level and/or domain-specific ontologies to form more coherent resources.

The other dimension along which ontologies can be categorized is by the richness of their internal structure. Lassila and McGuinness [2001] define categorization as a linear spectrum, as shown in Figure 3.4, that goes from the most simple ontologies to the most complex and powerful. Whether or not all of these definitions would actually be accepted as ontologies, depends on how one defines ontologies. Lassila and McGuiness divide the spectrum into two partitions, indicated by an inclining line, that define whether or not the definitions have a strict subclass hierarchy.



What is an Ontology?

Catalog/ ID

Thesauri “narrower term” relation

Formal is-a

Frames (properties)

General Logical constraints

Terms/ glossary

Informal is-a

Formal instance

Value Restrs.

Disjointness, Inverse, part-of…

Figure 3.4: *Lassila and McGuinness categorization of ontologies*

Correspondingly, with respect to richness, Corcho et al. [2003] introduce a division between lightweight and heavyweight ontologies. Lightweight ontologies include concepts, concept taxonomies, relationships between concepts, and properties that describe concepts, whereas heavyweight ontologies are lightweight ontologies with axioms and constraints added.

3.4 Representation Formalisms

Ontologies are special kinds of knowledge resources, ranging from simple concept taxonomies, like the hierarchies of domains available in search engines such as Altavista and Google and the hierarchies of topics for bibliographic categorization, to complex ontologies embedded in formal systems with reasoning capabilities allowing for new knowledge to be deduced from existing, automatic classification, as well as for validating knowledge through consistency checks.

In this thesis, the main purpose of introducing ontologies is to move from a query evaluation based on words to an evaluation based on concepts, thus moving from a lexical to a semantic interpretation. The goal is to use the knowledge in the ontologies to match objects and queries on a semantic basis.

For this purpose, it is necessary to be able to derive any similarity between concepts in the ontology, e.g. how closely related are *dogs* and *cats*, and are they more closely related than *dogs* and *postmen*? For a query on *dogs*, which of the two, *cats* or *postmen*, would be the most relevant answer? Of course, there is no obvious answer to this question because it depends on the context.

The representation of knowledge, an important research issue in the field of artificial intelligence (AI), has been researched and discussed since the beginning of AI. A lot of different approaches have evolved, and so a key question is which of these existing approaches is most suitable for a given task, which in this case is query evaluation involving conceptual similarity in the area of information retrieval. One could argue that before this selection is achievable, an attempt should be made to clarify exactly what knowledge representation is. This is precisely what Davis, Shrobe, and Szolovits do in [Davis *et al.*, 1993] and they conclude that:

1. **A knowledge representation is a surrogate**. Most of the things that we want to represent cannot be stored in a computer, e.g. bicycles, birthdays, motherhood, etc., so instead, symbols are used as a surrogate for the actual objects or concept. Perfect fidelity to the surrogate is impossible[2], and not necessary most of the time, as in most cases simple descriptors will do.

2. **A knowledge representation is a set of ontological commitments**. Representations are imperfect approximations of the world, each attending to some things and ignoring others. A selection of representation is therefore also a decision about how and what to see in the world. This selection is called the ontological commitment, i.e. the glasses that determine what we see.

3. **A knowledge representation is a fragmentary theory of intelligent reasoning**. To be able to reason about the things represented, the representation should also describe their behavior and intentions. While the ontological commitment defines how to see, the recommended inferences suggest how to reason.

4. **A knowledge representation is a medium for efficient computation**. Besides guidelines on how to view the world and how to reason, some remarks on useful ways to organize information are given.

5. **A knowledge representation is a medium of human expression**. The knowledge representation language should facilitate communication.

[2] It is not possible because anything other than the thing itself is necessarily different.

Davis et al. argue that among different achievements to be attained by understanding and using this description, one is, what they call, a characterization of the spirit of a given representation. In other words, all representations can more or less be seen as some fragment of first order logic, and can, based on this point of view, be equal or interchangeable, even if this sometimes requires extensive stretching and bending, though the intentions for the representations, or the spirit, could be very different. The selection of representations should therefore be on the basis of spirit, and the representations with intentions similar to the given task should be preferred.

An ontology is essentially a set of concepts connected by a set of relations, usually with subsumption/concept inclusion as ordering relation, forming the hierarchy of the ontology. Any ontology representation should at least accomplish this to be considered as a possible solution. Two of the most dominant models are network-based and logical representations. As mentioned previously, specific models in both of these categories could, for the most part, be translated into first-order logic, but their intentions differ. While network-based representations have their offspring in cognitive science and psychology, logical models are anchored in philosophy and mathematics.

Two of the most applied network-based models are *semantic networks* [Quillian, 1985] and *frames* [Minsky, 1985]. Although *semantic networks* and *frames* are significantly different, they share cognitive intuition, their features, and their network structure. Network-based representation models have a human-centered origin and match well with our intuition about how to structure the world.

The *semantic network model* was proposed by Quillian in the late 1960s as a model to capture the way humans organize the semantics of words. In the semantic network model, objects, concepts, situations and actions are represented by nodes and the relationships between nodes are represented by labeled arcs. The labels in the nodes gain their meaning from their connections [Brachman, 1985]. One of the advantages of this model is its simplicity, which is also its weakness. For modeling simple things, the semantics of the model is clear, but whenever the model becomes more complex, the semantics of the links between nodes in the network are unclear and need an explicit definition to be understood correctly [Woods, 1985]. Different representations are therefore not necessary, comparable, due to this vague semantic of relations, to the semantic network model.

The networks created by the semantic network model could also be created by the *frames model*, but instead of having simple nodes, the frames model is more object-centered and has nodes which are defined as a set of named slots. In psychology, a frame, or a scheme, is a mental structure that represents some aspect of the world, and is used to organize current knowledge and provide a framework for future understanding. This description is in fact indistinguishable from one describing how knowledge in artificial intelligence should be used and is thus also the inspiration for the frames model, which was proposed by Minsky in 1975.

Frames are pieces of knowledge, non-atomic descriptions with some complexity that are not necessarily complete with regard to details, but that are adequate for a given purpose. If one remembers a dog from the past, the shape of the nails on the left front paw may not be significant, but the color, nose, eyes, and overall shape might be. Frames are the structures to be filled with information about the piece of knowledge to be captured. Hence, a frame is a representational object with a set of attribute-value pairs defining the object. The specification of attributes typically includes [Levesque and Brachman, 1985]:

- Values defining an exact value for an attribute, or a default, to be derived from the structure,
- Restrictions defining constraints for the attribute's values, which could be value restrictions or number restrictions,
- Attached procedures providing procedural advice on how the attribute should be used and how to calculate the value (derive it from the structure), or they trigger what to do when the value is added,

where both the ability to define default values and to include procedural information about reasoning differs significantly from the semantic network model.

Frames are divided into generic types defining stereotyped situations and instances representing actual objects in the world. The generic types define the foundations, i.e. which slots define a specific type and perhaps the restrictions on the values of one or more slots, while instances fill out these slots with the values defining the instance.

Assuming that a stereotypical dog is characterized by being a specialization of the super-type *animal*, having a *color*, a *size*, and *number-of-legs*, as shown in Figure 3.5, notated as a attribute-value matrix:

$$\begin{bmatrix} \textbf{dog} \\ \begin{bmatrix} \textsc{supertype} & \text{animal} \\ \textsc{color} & \text{(type color)} \\ \textsc{size} & \text{(type size) (one of } \textit{small, medium, big}\text{)} \\ \textsc{number-of-legs} & 4 \end{bmatrix} \end{bmatrix}$$

Figure 3.5: *An example of the stereotype dog notated in a attribute-value matrix*

where the value of the attributes *color* and *size* are restricted to a specific type, where the attribute *size*, furthermore, is restricted to a set of possible values, and the attribute *number-of-legs* is fixed to the value four.

In Figure 3.6, an instance *fido* of the generic frame *dog* is shown. The values of the attributes *color* and *size* are fixed, and thus all attributes of the instance are fixed.

$$\begin{bmatrix} \textbf{fido} \\ \begin{bmatrix} \textsc{super-type} & \text{dog} \\ \textsc{color} & \text{gray} \\ \textsc{size} & \text{big} \end{bmatrix} \end{bmatrix}$$

Figure 3.6: *An example of a frame denoting the instance fido of the stereotype dog*

The instance, *fido*, inherits all attributes from both of its parents, thus even though the *number-of-legs* attribute is not shown in Figure 3.6, it is inherited from the super-type *dog*.

Frames were a step ahead of semantic networks, but still the semantics of relations were unclear, links could represent implementational pointers, logical relations, semantic relations, and arbitrary conceptual and linguistic relations [Brachman and Schmolze, 1985]. One attempt to bring more strict semantics into frames and semantic networks was done in the representation language KL-ONE [Brachman and Schmolze, 1985], where there were a clear distinction made between different kinds of links, which has been the main criticism against semantic networks [McDermott, 1976; Woods, 1985]. In Figure 3.7, the example with the type *dog* and the instance *fido* is shown in KL-ONE. The ovals define concepts and the arrows different kinds of relations. The double-line arrows represent the subtype-super-type relations, and the arrows with the encircled square represent roles. The v/r at the target end of the role arrows indicates value restrictions or type constraints.

In KL-ONE and other frame-based knowledge representation languages, knowledge is split into two different parts; the terminological part denoted the TBox, and the assertional part, denoted the ABox. The TBox concerns taxonomies of structured terms and the ABox descriptive theories of domains of interest [Brachman *et al.*, 1985].

Even though the semantics of relations is strict in KL-ONE, it can become difficult to give a precise characterization of what kind of relationship can be computed when more complex relationships are established among concepts. This problem and the recognition that the core features of frames could be expressed in first order logic [Hayes, 1985], were among the main motivations for the development of *description logic* [Nardi and Brachman, 2003]. It was shown that only a fragment of first order logic was needed, and that different

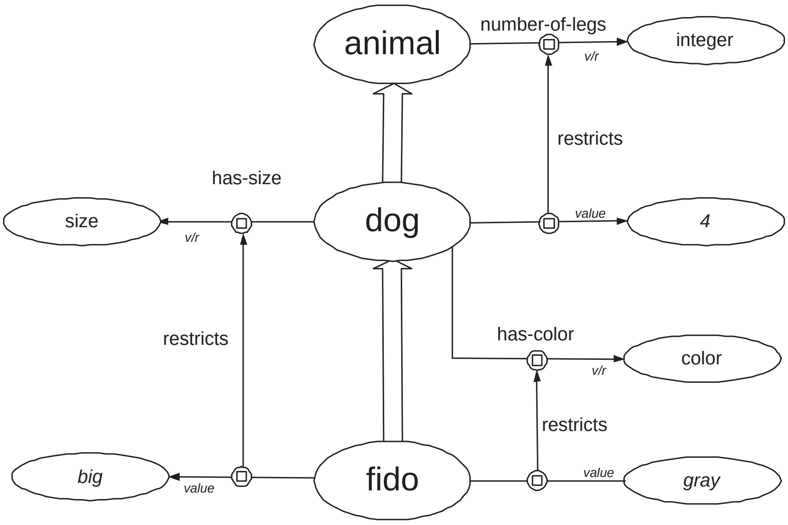


Figure 3.7: *The example with the type dog and the instance fido in KL-ONE*

features of the representation language lead to different fragments of first order logic. This was also true for the reasoning part, which meant that it did not necessarily require proof of full first-order logic theorems, and that the reasoning in different fragments of first-order logics would lead to computational problems of differing complexity.

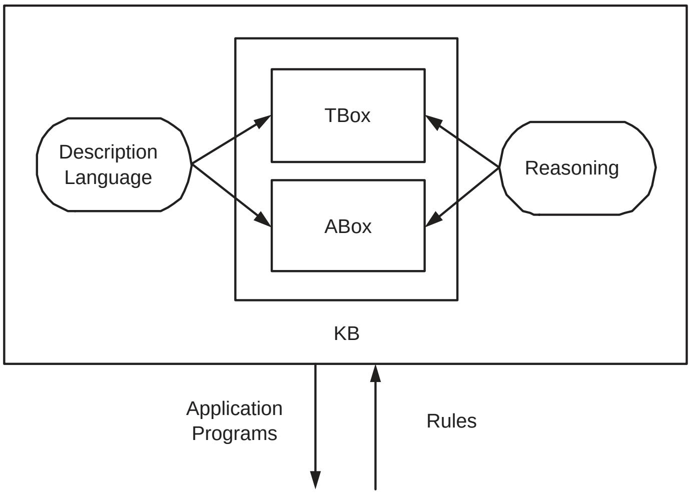


Figure 3.8: *Architecture of the knowledge representation system in description logic ([Baader and Nutt, 2003])*

Description logics inherit the division of knowledge representation into a terminological and an assertional part, as shown in Figure 3.8. As mentioned above, description logics are a set of languages that are all a subset of first-order logic. All the languages have in common elementary descriptions that are atomic concepts and atomic roles. Complex descriptions can be built from the atomic ones inductively with concept constructors, and each language is therefore distinguished by the constructors they provide. As a minimal language of practical interest, the attributive language $\mathcal{AL}$ was introduced in [Schmidt-Schaubss and Smolka, 1991]. In Figure 3.9, the syntax rule for the $\mathcal{AL}$ language, defined in an abstract notation where the letters A and B denote atomic concepts, R atomic roles, and C and D concept descriptions.

$$\begin{array}{rll}
C, D \rightarrow & A| & \text{(atomic concept)} \\
& \top| & \text{(universal concept)} \\
& \bot| & \text{(bottom concept)} \\
& \neg A| & \text{(atomic negation)} \\
& \sqcap D| & \text{(intersection)} \\
& \forall R.C| & \text{(value restriction)} \\
& \exists R.\top & \text{(limited existential quantification)}
\end{array}$$

Figure 3.9: *The basic description language* $\mathcal{AL}$

Whenever the $\mathcal{AL}$ language is extended, the extension is denoted by adding letters to the language name, for instance the number restriction is denoted by the letter $\mathcal{N}$, hence the $\mathcal{ALN}$ language is the $\mathcal{AL}$ language with number restrictions. In order to express the example in Figure 3.5 and 3.7 in description logics, the $\mathcal{ALN}$ language is needed. The result is shown in Figure 3.10, where the double vertical line divides the example in an upper TBox and a lower ABox.

Although, the different knowledge representation models described up to this point are closely related, they relate to at least two different kinds of spirits, or ontological commitments, the cognitive and the logical. A third spirit, the mathematical or algebraic spirit, will now be introduced. [Brink *et al.*, 1994] shows that the *Peirce Algebra* and the description language $\mathcal{U}^-$ are isomorphic, where the $\mathcal{U}$ language is one of the most powerful languages having $\mathcal{ALC}$ language[3] as a sub-language, and the $\mathcal{U}^-$ language is the $\mathcal{U}$ language without the numerical restrictions; *atleast* and *atmost.*

The model-theoretic semantics of a description language can be given by an interpretation $\mathcal{I}$, which defines a pair $(\mathcal{D}^{\mathcal{I}}, .^{\mathcal{I}})$, where $\mathcal{D}^{\mathcal{I}}$ denotes a set called the domain or universe, and $.^{\mathcal{I}}$ is a map, called the interpretation function, which assigns to every concept description C a subset $C^{\mathcal{I}}$ of $D^{\mathcal{I}}$ and to every

[3] The $\mathcal{ALC}$ language or the $\mathcal{ALUE}$ language is $\mathcal{AL}$ language plus either full negation or union and full existential quantification, which, of course, defines the same language. $\mathcal{ALC}$ is normally used to denote it.

$$\begin{array}{lcl} \text{Dog} & \equiv & \text{Animal} \sqcap \\ & & \leqslant \text{numberOfLegs}(4) \sqcap \\ & & \geqslant \text{numberOfLegs}(4) \sqcap \\ & & \forall \text{hasColor}(\text{Color}) \sqcap \\ & & \forall \text{hasSize}(\text{Size}) \end{array}$$

Dog(fido)
hasSize(fido, big)
hasColor(fido, gray)
Color(gray)
Size(small)
Size(medium)
Size(big)

Figure 3.10: *Example of dog with the instance fido in description logic*

role R, a binary relation $R^{\mathcal{I}}$ over $D^{\mathcal{I}}$. Figure 3.11 shows the conditions for a subset of the $\mathcal{U}$-language and the mapping to the matching algebraic terms:

Terminological Expression	Interpretation	Algebraic Term
$\top$	$\mathcal{D}^{\mathcal{I}}$	1
$\bot$	$\emptyset$	0
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$a \cdot b$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	$a + b$
$\neg C$	$\mathcal{D}^{\mathcal{I}} - C^{\mathcal{I}}$	a'
$\exists R.C$	$\{x \mid (\exists y)[(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}]\}$	$r : a$
$\forall R.C$	$\{x \mid (\forall y)[(x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}]\}$	$(r : a')'$

Figure 3.11: *A logical connection between Peircian algebra and the description language $R^{\mathcal{U}}$*

where $r : a$ denotes Peirce's product [Brink, 1981], where r is an element in a relation algebra and a, b are elements in a Boolean algebra.

The example in Figure 3.10 could obviously be represented in Peircian algebra, except for the number restriction, so unless cardinality is essential for a given representation task, Peircian algebra would be a possible representation model with a mathematical or algebraic spirit.

The goal here has not been to give an overview of representation models, but to show the logical similarities and differences of the spirit and intentions of the models presented. Nevertheless, these models give a representative glimpse of the evolution towards a well-defined semantics of relations between elements of knowledge.

Selecting which language is the most appropriate for representing a given ontology, first of all, depends on the type and richness of the ontology, as well as on the type of reasoning needed. A "lowest common denominator" view of the notion of an ontology is indicated in [Uschold and Jasper, 1999]:

> *An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms, and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.*

Thus, the representation could be rather simple – a mapping from the vocabulary to the concepts of the ontology (with specification of meaning), and a specification of one or more relations of the concepts. As a result, a representation could be fairly straightforward and simple. However, when the need for richness increases, moving from left to right in Figure 3.4, the choice of representation languages and formalisms become less straightforward.

For simple representations of ontologies without the need for reasoning, any language with the ability to describe links between entities would be appropriate, i.e. Unified Modeling Language (UML) [Object Management Group,], Entity-Relationship Model (ER) [Chen, 1976], or other software engineering languages. Whenever reasoning becomes decisive, the differences and similarities among formalisms described in the last section should be taken into account when choosing a representation language. Gómez-Pérez et al. [2004] conclude that both the formalism used to model the ontology and the language that implements these techniques limit the kind of knowledge that can be modeled and implemented.

Other aspects which should be taken into consideration concern the type of information to be modeled and the applicability. Examples of different types of knowledge are top-level knowledge, common knowledge, domain-specific knowledge, while examples of applicability are data integration and ontology modeling tools.

3.4.1 Traditional Ontology Languages

In the beginning of 1990s, a set of Artificial Intelligence representation languages for ontologies was created. Some were based on first order logic, primarily based on the Knowledge Interchange Format (KIF) [Genesereth, 1991], while others were based on frames combined with first order logic, for instance, the language Ontolingua [Gruber, 1992]. Ontolingua, which is one of the most expressive of all the languages that have been used for representing ontologies, allows the representation of concepts, taxonomies of concepts, n-ary relations, functions, axioms, instances, and procedures. Because its high expressiveness led to difficulties in building reasoning mechanisms for it, no reasoning support is provided with the language [Corcho *et al.*, 2003].

Later (1995), the language FLogic [Kifer *et al.*, 1995], which also combines frames and first order logic, was created. FLogic allows the representing of concepts, concept taxonomies, binary relations, functions, instances, axioms, and deductive rules. The FLogic language has reasoning mechanisms for constraint checking and the deduction of new information.

The language LOOM [MacGregor, 1991], which was developed simultaneously with the Ontolingua, is a general knowledge representation language based on description logic and production rules and provides the automatic classification of concepts. In addition, ontologies can be represented with concepts, concept taxonomies, n-ary relations, functions, axioms and production rules.

3.4.2 Ontology Markup Languages

The Internet boom has lead to a number of web-based representation languages. Introduced in 1996, the first one, SHOE [Heflin *et al.*, 1999], was an extension of HTML and introduced tags, which allow the insertion of ontologies in HTML documents. Later, when XML [Paoli *et al.*, 2004] was created and widely adapted, SHOE was modified to use XML. SHOE combines frames and rules, and allows representation for concepts, their taxonomies, n-ary relations, instances and deduction rules that can be used by its inference engine to obtain new knowledge.

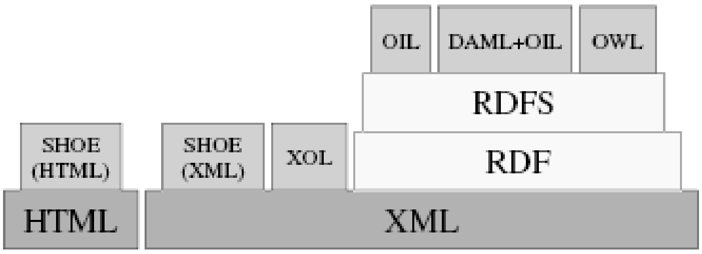


Figure 3.12: *Ontology markup languages*

The release and use of XML gave rise to a large number of different ontology markup languages, as shown in Figure 3.12, where the languages based on the Resource Description Framework (RDF) [Lassila and Swick, 1999] are based on description logic. The RDF language is in itself very limited and primarily designed for describing web resources, but the extension RDF Schema (RDFS) [Brickley and Guha, 2004] added frame-based primitives. These did not make

RDFS very expressive, since they only allow the representation of concepts, concept taxonomies and binary relations. Inference engines have been created for this language, mainly for constraint checking. Ontology Inference Layer or Ontology Interchange Language (OIL) [Fensel *et al.*, 2000] was developed in the framework of the European IST project, On-To-Knowledge, which adds frame-based knowledge representation primitives to RDF(S). Its formal semantics are based on description logic, and it supports the automatic classifications of concepts. DAML+OIL (DARPA agent markup language) [Connolly *et al.*, 2001], which is a union of DAML with OIL, also adds description logic primitives to RDF(S), and allows representing concepts, taxonomies, binary relations, functions and instances. The OWL Web Ontology Language [Bechhofer *et al.*, 2004], which is the latest branch of ontology markup languages, covers most of the features in DAML+OIL. The differences between OWL and DAML + OIL are mainly a change of name of the original DAML+OIL primitives, since they were not always easy to understand for non-experts.

The traditional representation languages presented are primarily based on first order logic and frames (except for LOOM), while the ontology markup languages primarily are based on descriptions logic. The ability of reasoning is a very important issue for these paradigms, and the development of fast model checkers for descriptions logic (see e.g. [Horrocks, 1998]) is a huge improvement on the applicability of descriptions logic in real-life applications. Still, reasoning in large ontologies is computationally complex and its usefulness, for instance, in large-scale information retrieval systems is limited.

The aim of this thesis is to introduce ontologies for large-scale information retrieval, thus reasoning is aimed at being avoided, and instead, the idea is to use measures of similarity between concepts derived from the structure of the ontology, and by doing so, replace reasoning over the ontology with numerical similarity computation. This is aimed at being achieved by transforming ontology representation into directed graphs in which well-known algorithms can be used to compute, for instance, the shortest path between nodes. The intrinsic graphical nature of the lattice-algebraic representation seems therefore to be an obvious choice.

3.4.3 Ontolog

The ontology sources to be used in this thesis are linguistic ontologies and the ontological framework characterized as a *generative ontology* [Andreasen and Nilsson, 2004] in the generative grammar tradition of Chomsky and with ties to the *generative lexicon semantics* found in [Pustejovsky, 1991]. A generative ontology is an ontology defined by a set of atomic concepts and a set of relations. From these two sets, assumed given by the ontological sources, any concept combining atomic or already combined concepts by relations is also a part of the generative ontology.

This generative ontological framework is modeled by the lattice-algebraic description language called ONTOLOG [Nilsson, 2001]. The primary role of the ontology is to provide a specification of a shared conceptualization in an (ontological) information retrieval environment targeting only partial representation of meaning. The aim is to capture and represent various aspects of meanings from both queries and documents and use the ontological framework to measure similarity.

Semantic Relations	
temporal aspect	TMP
location, position	LOC
purpose, function	PRP
with respect to	WRT
characteristic	CHR
with, accompanying	CUM
by means of, instrument, via	BMO
caused by	CBY
causes	CAU
comprising, has part	CMP
part of	POF
agent of act or process	ACT
patient of act or process	PNT
source of act or process	SRC
result of act or process	RST
destination of moving process	DST

Table 3.2: *The subset of semantic relations R and their abbreviations as presented in [Nilsson, 2001]*

Semantic relations are used for combining concepts, expressing feature attachment, and thereby forming compound concepts. The number and types of relations in the ontology source and in the query language should be harmonized and reflect the granularity of relations in use. There have been numerous studies regarding representative collections of semantic relations. The set of relations presented in Table 3.2 constitutes a limited selection that can represent "meaning" on a very coarse level for a subset of the occurring concepts.

Terms in the ONTOLOG representation are well-formed concepts situated in an ontology, with concept inclusion as the key ordering relation.

The basic elements in ONTOLOG are concepts and binary relations between concepts. The algebra introduces two closed operations on the concept expressions φ and ψ [Nilsson, 2001]:

- conceptual *sum* $(\varphi + \psi)$, interpreted as the concept being φ or ψ
- conceptual *product* $(\varphi \times \psi)$, interpreted as the concept being φ and ψ

Relationships r are introduced algebraically by means of a binary operator (:), known as the Peirce product $(r : \varphi)$, which combines a relation r with an expression φ, resulting in an expression, thus relating nodes in the ontology.

The Peirce product is used as a factor in conceptual products, as in $c \times (r\colon c_1)$, which can be rewritten to form the feature structure $c[r\colon c_1]$, where $[r\colon c_1]$ is an attribution of the concept c.

Thus, the attribution of concepts to form compound concepts corresponds to feature attachment. Attribution of a concept a with relation r and concept b has the intuitive lattice-algebraic understanding $a \times r(b)$, conventionally written as the feature structure $a[\textsc{r}{:}b]$.

Algebraically, the concept $a[\textsc{r}{:}b]$ is therefore to be understood as the greatest lower bound (meet, infimum) for all as and all concepts being r-attributed with the value b.

Given atomic concepts $\mathbf{A}$ and relations $\mathbf{R}$, the set of well-formed terms $\mathbf{L}$ of the Ontolog language is defined as follows:

- if $x \in \mathbf{A}$ then $x \in \mathbf{L}$,
- if $x \in \mathbf{L}$, $r_i \in \mathbf{R}$ and $y_i \in \mathbf{L}, i = 1, \ldots, n$ then $x[r_1\colon y_1, \ldots, r_n\colon y_n] \in \mathbf{L}$.

Compound concepts can thus have multiple and nested attributions. Examples of such compound concepts are the concepts $dog[\textsc{chr}{:}gray[\textsc{chr}{:}dark]]$, and $dog[\textsc{chr}{:}black, \textsc{chr}{:}fast]$, respectively. The attributes of a compound concept $X = x[r_1\colon y_1, \ldots, r_n\colon y_n]$ are considered as a set, and thus X can be rewritten with any permutation of $\{r_1\colon y_1, \ldots, r_n\colon y_n\}$ [Andreasen *et al.*, 2003c].

3.5 Resources

In this section, a number of resources will be presented that can act as a basis for the ontological framework just presented. Normally, not just a single source is used as the foundation, but a number of different sources are merged to form the basis for a generative ontology framework.

The resources presented here are just a small fragment of the possible ontological resources available. The large lexical database for English, WordNet [Miller *et al.*, 1990; Miller, 1995], was chosen because it is an extensive framework used in many projects related to ontologies.

In the OntoQuery project, a small Danish ontology SIMPLE [Pedersen and Keson, 1999; Pedersen and Nimb, 2000] is used in connection with Danish language texts. Although not a perfect resource due to certain defects (see e.g. [Bulskov and Thomsen, 2005]), it is the only ontology of a reasonable size available in Danish. Fortunately, a newly started project, DanNet, has the goal of creating a Danish version of WordNet. This, of course, substantiates the use of WordNet, since the Danish version later can be merged into the framework discussed here.

Before WordNet is presented in Section 3.5.3, two examples of using WordNet in combination with a top-ontology are presented as well as a top-ontology mergeable with WordNet and a complete ontology created by merging different sources (including WordNet).

3.5.1 Suggested Upper Merged Ontology

The Standard Upper Ontology (SUO) provides definitions for general-purpose terms and acts as a foundation for more specific ontologies [Pease *et al.*, 2001]. The Suggested Upper Merged Ontology (SUMO) was created with input from SUO and by merging publically available ontological content into a single, comprehensive and cohesive structure. SUMO is being created as part of the Institute for Electrical and Electronics Engineers' (IEEE) Standard Upper Ontology Working Group [SUMO, 2006], whose goal is to develop a standard upper ontology that will promote data interoperability, information search and retrieval, automated inference, and natural language processing. The goal of SUMO, which is represented in KIF [Genesereth, 1991], is to create an upper ontology by merging well-known upper ontologies and knowledge resources, for instance, the Ontolingua server [Gruber, 1992], Sowa's upper level ontology (see Section 3.2), Allen's temporal axioms [Allen, 1984], plan and process theories [Pease and Carrico, 1997], and various mereotopological theories [Borgo *et al.*, 1996a; Borgo *et al.*, 1996b].

The introduction of an upper ontology has several advantages. For example, it can serve as a basis for knowledge integration and translation between ontologies, as well as tie together domain specific ontologies.

The noun portion of WordNet, version 2.1, is currently mapped to SUMO [Niles and Pease, 2003]. This mapping of SUMO and WordNet can serve as a foundation for combining WordNet with other resources, with SUMO acting as a bridge between the lower level ontologies. Furthermore, the mappings between WordNet and SUMO can be regarded as a natural language index to SUMO.

Sometimes the "conceptual distance" between the top-level ontology and the domain specific ontologies is too wide, and so-called mid-level ontologies can be used to close the gap. The mid-level ontology, MILO [Niles and Terry, 2004], is intended to act as such a bridge between the high-level abstractions of SUMO and the low-level detail of domain ontologies, for instance, between SUMO and WordNet.

3.5.2 SENSUS

SENSUS [Knight and Luk, 1994] is a natural language-based ontology developed by the Natural Language Group [NLG, 2006] at Information Sciences Institute (ISI) to provide a broad conceptual structure for working in machine translation.

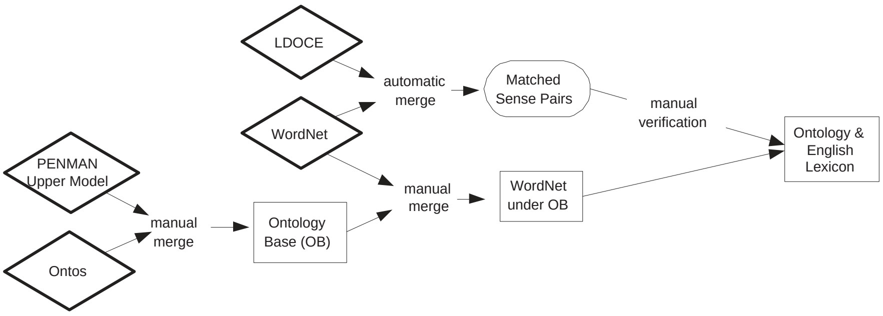


Figure 3.13: *Merging information in four linguistic resources into the ontology SENSUS*

SENSUS contains more that 50,000 nodes representing commonly encountered objects, entities, qualities, and relations, and was built using of a number of different linguistic resources, including: the PENMAN upper model [Bateman *et al.*, 1989; Bateman, 1990]; ONTOS top-level ontology [Carlson and Nirenburg, 1990]; Longman's Dictionary of Contemporary English (LDOCE); and WordNet [Miller *et al.*, 1990][4].

Figure 3.13 shows the process of merging the information in the four resources into the SENSUS ontology. The PENMAN Upper Model and ONTOS were merged manually into the Ontology Base (OB). WordNet was then subordinated (merged) into the OB, resulting in a large knowledge base. The next phase was to merge LDOCE and WordNet. The motivation for this was that LDOCE has lexical information not present in WordNet[5].

Originally, SENSUS was solely intended as a resource for machine translation, but with more than 50,000 nodes, it also appeared to be useful as an ontology for other knowledge-based systems, such as the one in focus here.

[4] SENSUS also uses Collins Bilingual Dictionary (Spanish-English) for machine translation, but this is not the issue here.

[5] Another reason was that LDOCE sense identifiers are legal tokens in the bilingual merge.

3.5.3 WordNet

WordNet is a large lexical database for English created at Princeton University [Miller *et al.*, 1990; Miller, 1995]. The unit of WordNet is words, as the name indicates, even though it contains compounds, phrasal verbs, collections, and idiomatic phrases. The main purpose of WordNet was to establish a lexical resource for psycholinguistics and computational linguistics of such a magnitude that huge amounts of lexical knowledge would be available as well as a structure that could support linguistic research better than traditional dictionaries.

The fundamental idea behind WordNet was to organize lexical information in terms of meanings, rather than word form, thus moving from a traditional dictionary towards a thesaurus by including semantic relations between words. Normally, the word "*word*" is used to refer to both the utterance and to its associated concept. Therefore, a distinction is made by using *word form* when referring to the physical utterance and *word meaning* when referring to the lexicalized concept that the form can be used to express, or by using simply *word* and *meaning* or *sense*.

In WordNet, word forms and word meanings are mapped similar to a dictionary. If one word form maps to more than one meaning, the form is polysemous, and if more than one form maps to the same meaning, these forms are synonymous. This is shown in the lexical matrix in Table 3.3, where the forms F_1 and F_2 are synonymous with respect to meaning M_1, while form F_2 is polysemous and hence maps to meanings M_1 and M_2.

Word	Word Forms				
Meanings	F_1	F_2	F_3	$\dots$	F_n
M_1	$\mathrm{E}_{1,1}$	$\mathrm{E}_{1,2}$			
M_2		$\mathrm{E}_{2,2}$			
M_3			$\mathrm{E}_{3,3}$		
$\vdots$				$\ddots$	
M_m					$\mathrm{E}_{m,n}$

Table 3.3: *Lexical matrix*

Traditional dictionaries are solely ordered alphabetically by word form. WordNet is also ordered in a taxonomical hierarchy of meanings, thus providing a significant contribution to an ontology. The concepts are specified by lexicalization, where synonymous word forms are grouped into sets defining the meaning they share. These synonym sets are called *synsets* and constitute the unit of meaning in WordNet. The understanding of synonymy in WordNet is that two word forms are synonymous in a linguistic context C if the substitution of one form for the other in C does not alter the truth value[6]. This restricts the word forms in a synset to one syntactic category, since forms from different categories will not be interchangeable based on the above definition of the synonym relation. For this reason, WordNet is divided into four separate semantic nets, one for each of the open part of speech classes: nouns, adjectives, adverbs and verbs.

There is a large variety of semantic relations that can be defined between word forms and between word meanings, but only a small subset of them is used in WordNet, because they apply broadly throughout the English language and because they are familiar to ordinary users who do not need special training in linguistics to understand them [Miller, 1995]. WordNet includes the following semantic relations:

- *Synonymy* is the basic relation in WordNet, because it is used to form sets of synonyms to represent word meanings. Synonymy is a symmetrical relation between word forms.

- *Antonymy* is also a symmetrical relation between word forms, especially important in organizing the meanings of adjectives and adverbs.

- *Hyponymy* and its inverse, *Hypernymy*, are the subordinate or specification relation and superordinate or generalization relation, respectively, as well as the transitive and asymmetric relations between word meanings, i.e. synsets, forming a hierarchy among nouns.

- *Meronymy* and its inverse, *Holonymy*, are the part-whole relations.

- *Troponymy* (manner-name) is for verbs what hyponymy is for nouns, although the resulting hierarchy is much shallower.

- *Entailment* is between verbs, e.g. snoring entails sleeping.

The database statistics for the current version, 2.1, of WordNet is shown in Table 3.4 [WordNet, 2005].

Nouns

Nouns form an integral part of WordNet. As shown in Table 3.4, more than 75% (117,097 out of 155,327) of the words are nouns. The semantic relation that organizes nouns into a lexical hierarchy is the hyponym relation (and its inverse relation hypernym), which relates the lexicalized concepts defined by synsets and forms the hierarchy. This relation can be read as the IS-A or

[6] This definition of synonymy restricts a relation to being valid in only "some" contexts, in contrast to the more strict definition, where the relation should be valid in any context, which would only give rise to very few synonyms in natural language.

POS	**Unique Strings**	**Synsets**	**Total Word-Sense Pairs**
Noun	117097	81426	145104
Verb	11488	13650	24890
Adjective	22141	18877	31302
Adverb	4601	3644	5720
Total	155327	117597	207016

Table 3.4: *Number of words, synsets, and senses in WordNet 2.1*

IS-A-KIND-OF relation, and can also be denoted as a subordinate relation, concept inclusion, or subsumption relation. The nouns in WordNet form a lexical inheritance system where properties are inherited downwards in the hierarchy through the hyponym relation. For instance, if the synset {*dog, domestic dog, Canis familiaris*} is a hyponym of the synset {*canine, canid*}, which has the properties "mammals with non-retractile claws and typically long muzzles", then it will be inherited by the synset {*dog, domestic dog, Canis familiaris*}.

Often the structure of the hierarchy of nouns is modeled as a single hierarchy with only one unique beginner, thus all concepts are assumed interconnected, in particular when use of a top-level ontology is assumed. Originally, this was not the case in WordNet, which had 25 unique beginners, as shown in Table 3.5, thus forming 25 individual hierarchies. These hierarchies vary in size and are not mutually exclusive, but on the whole they cover distinct conceptual and lexical domains [Miller, 1998]. In the current version of WordNet (version 2.1) the 25 individual hierarchies is merged into a single hierarchy with one unique identifier, the {*entity*} synset. SUMO and SENSUS are examples of how one can merge, for example these 25 hierarchies, into a single interconnected ontology, while this is not how the merging is done in WordNet.

{*act, activity*}
{*animal, fauna*}
{*artifact*}
{*attribute*}
{*body*}
{*cognition, knowledge*}
{*communication*}
{*event, happening*}
{*feeling, emotion*}
{*food*}
{*group, grouping*}
{*location*}
{*motivation, motive*}
{*natural object*}
{*natural phenomenon*}
{*person, human being*}
{*plant, flora*}
{*possesssion*}
{*process*}
{*quantity, amount*}
{*relation*}
{*shape*}
{*state*}
{*substance*}
{*time*}

Table 3.5: *List of the 25 unique beginners in WordNet*

Adjectives and Adverbs

Adjectives in WordNet are divided into two categories: descriptive and relational adjectives. The descriptive adjectives are the kind of adjectives that usually comes to mind when adjectives are mentioned, e.g. “small”, “beautiful”, “possible”, etc. Relational adjectives are called relational simply because they are related by their derivation to nouns. Neither of these categories forms a hierarchy like the hyponymy hierarchy for nouns, due to the fact that the hyponymy relation does not hold for adjectives; it is not clear what it means to say that one adjective “is a kind of” of another adjective [Fellbaum, 1998]. The basic semantic relation for adjectives in WordNet is the antonymy relation. The reason for this results from word association tests, which show that when the probe is a familiar adjective, the response commonly given by adult speakers is the antonym [Fellbaum, 1998].

Descriptive adjectives ascribe to a noun a value of an attribute. To say “*the car is fast*” presupposes that the noun “*car*” has an attribute “*speed*” which can take the value “*fast*”. Antonymous adjectives express opposite values of an attribute, and in WordNet, these opposite pairs (antonyms) of adjectives are used to form a cluster. Not all the descriptive adjectives have antonyms, and will therefore not be able to form a cluster. Instead, a similarity relation is introduced to indicate that the adjectives lacking antonyms are similar in meaning to adjectives that do have antonyms. The similarity relation between adjectives is a sort of specialization. It states that if adjective x is similar to adjective y, then the class of nouns that can be modified by x is a subset of the class of nouns that can be modified of y. This structure forms clusters with the antonym pair as heads and all the similar adjectives as satellites to one of the heads, as shown in Figure 3.14. Taking Figure 3.14 as an example, the adjective “*humid*” is a satellite (similar) to the adjective “*wet*”, which would then mean that the class of nouns that can be modified by “*humid*” is less or equal to the class of nouns that can be modified by “*wet*”. All the descriptive adjectives have an antonym, either directly or indirectly through their similar relation to other adjectives.

The other class of adjectives, the relative adjectives, consists of all the adjectives related, semantically or morphologically, to nouns. They do not form an independent structure like the clusters of descriptive adjectives, but are connected to the nouns from which they are derived. One of the main differences between relative and descriptive adjectives is that relative adjectives do not refer to a property of the noun they modify, hence they do not relate to an attribute. A small set of adjectives are derived from verbs, e.g. elapsed, married, boiling, etc., and have a relation to the verbs from which they are derived. Some of the relative adjectives have a direct antonym, e.g. married/unmarried, physical/mental, etc., and thus fit into the structure of

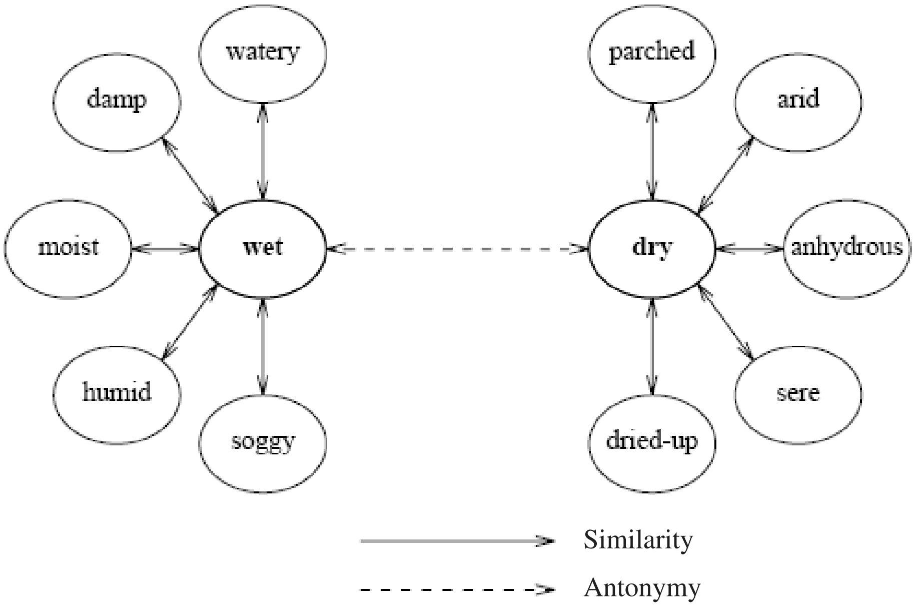


Figure 3.14: *Adjective antonym cluster with "wet" and "dry" as heads*

clusters and will therefore be organized into clusters as well.

In summary, the structure of adjectives in WordNet is divided up as follows: All descriptive adjectives with antonyms are organized in bipolar clusters; descriptive adjectives without antonyms are related to similar adjectives which have antonyms; and all relative adjectives have some kind of "*derived from*" relation to their source.

The semantic organization of adverbs in WordNet is simple and straightforward. There is no hierarchical structure for adverbs like there is for nouns and verbs, and no cluster structure like there is for adjectives. Because most adverbs are derived from adjectives by suffixation, they only have a derived from relation to the adjective origin. For some adverbs, synonymy and antonymy are recognized.

Statistical Information

The following two statistical measures in WordNet are of great importance to the tasks in this thesis: the frequency of senses and the familiarity of words. In WordNet, senses have statistical information attached about occurrences determined by frequency of use in semantically tagged corpora [Landes *et al.*, 1998]. This kind of information is very important in the process of disambiguating word senses, but can not solely determine the correct sense of the word in all cases, a subject further discussed in Section 4.2.2. In [Miller *et al.*, 1994], a comparative evaluation of word sense disambiguation by guessing, using frequencies and co-occurrence, is performed. The results are shown in Table 3.6 and indicate a significant difference when using sense frequencies compared to guessing.

Heuristic	Monosemous and Polysemous	Polysemous only
Guessing	45.0	26.8
Most Frequent	69.0	58.2
Co-occurrence	68.6	57.7

Table 3.6: *Percentage of correct sense identification for open-class words with and without information on sense frequencies*

Words are not used with the same frequency in texts; some words appear in almost all texts, e.g. “the”, “is”, “at”, “of”, and some words appear only rarely in specific domains. The frequency of use reflects the familiarity of words and hence how commonly well-known they are. Familiarity is known to influence a wide range of performance variables: speed of reading, speed of comprehension, ease of recall, probability of use. Frequency of use is usually assumed to be the best indicator of familiarity. However, the frequencies that are readily available in semantically tagged corpora are inadequate for a database as extensive as WordNet.

Fortunately, an alternative indicator of familiarity has been developed as it is known that frequency of occurrence and polysemy are correlated[0] [Jastrezembski, 1981]. The idea is that, on average, the more frequently a word is used, the more different meanings it will have in a dictionary, which means that a dictionary can be used to determine the familiarity of words instead of, e.g. corpora. Words in WordNet have attached a measure of familiarity similar to the number of different meanings in the dictionary, where 0 means not in the dictionary.

One use of familiarity is for visualizing. Visualization can be modified by hiding senses with low familiarity, thereby reducing long paths to produce a clearer picture when senses do not contribute significant information to comprehension.

Consider the example in Table 3.7, where the hyponyms of the sense “*bronco*” are shown. One could chose to hide all the senses with, for instance, a familiarity ≤ 1 (except for “*bronco*” of course), since these would probably not be significant to normal users, thereby making this path easier to visualize and comprehend. (see Section 6.5 for more on the visualization of ontologies).

Word	Polysemy
bronco	1
@ → mustang	1
@ → pony	5
@ → horse	14
@ → equine	0
@ → odd-toed ungulate	0
@ → placental mammal	0
@ → mammal	1
@ → vertebrate	1
@ → chordate	1
@ → animal	4
@ → organism	2
@ → entity	3

Table 3.7: *Hypernyms, denoted @ →, of "bronco" and their familiarity values*

3.5.4 Modeling Ontology Resources

The use of lexical resources, like WordNet, as linguistic ontologies seems to be continually growing. Because of the linguistic and relatively informal nature of linguistic ontologies such as WordNet, various problems are encountered when switching to the formal framework [Bulskov and Thomsen, 2005; Haav and Lubi, 2001; Goerz *et al.*, 2003].

One group of problems concern inconsistencies in the coding of relations to other concepts, e.g. circularity in the taxonomy or the inheritance of inconsistent features. Other problems are due to incomplete data, e.g. a lack of differentiating features on concepts. A third group of problems concerns the violation of ontological principles.

The first group of problems are trivial, but can be rather time consuming to solve in extensive resources.

The second group is bound to the limited set of semantic relations normally found in lexical resources - they mainly contribute taxonomic knowledge. Differentiating features on concepts are essential in the area of terminology and require a rather complex set of relations (see e.g. [Madsen *et al.*, 2001]), and will therefore seldom be present in large-scale linguistic ontologies. In Bulskov and Thomsen [2005] methods on this subject are discussed.

"*laundry*" is an example taken from WordNet that is coded with "*workplace*" as parent. "*workplace*" is furthermore parent for a rather large variety of concepts, e.g. "*farm*", "*bakery*", and "*shipyard*", etc., classified only by the relation to a common parent, and the description in the WordNet gloss. These concepts will be treated as atomic concepts when WordNet is used, even though they are compound, e.g. "*laundry*" can be described more correctly (or synonymously) as *workplace*[WRT:*laundering*]. This type of knowledge can be used in the detection of paraphrases, for instance, in a phrase like "*a workplace for laundering*" which could be described as *workplace*[WRT:*laundering*] and then be mapped to the "*laundry*" concept (or the extended {*laundry*, *workplace*[WRT:*laundering*]} synset).

Similarly, a rather large amount of words in WordNet are multi-word terms, e.g. "*private school*", "*general election*", "*rocket launching*". As demonstrated later (see 5), we aim for a similarity measure that takes into account not only the concept inclusion relation, but all kinds of semantic relations. In WordNet the only relation from, for instance, "*general election*" is to its parent "*election*". The form *election*[CHR:*general*] would in addition refer to "*general*". Concepts (synsets in WordNet) would then also be related by other relations, as shown in Figure 3.15, where a "*general election*" and a "*general manager*" are related as "*general things*".

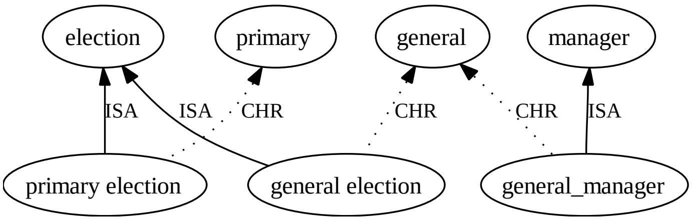


Figure 3.15: *A small fragment of an ontology showing the interpretations of the multiwords election*[CHR*:general*]*, election*[CHR*:primary*]*, and manager*[CHR*:general*]

The third group concerns the violation of ontological principles. Gangemi et al. [2003] list a number of main problems found in WordNet. There is confusion between concepts and individuals caused by the lack of an "instance of" relation, since both are related by the hyponym relation. They state that in the presence of an "instance of" relation, they would be able to distinguish between concept-concept relations and individual-concept relations (see Gangemi et al. [2003] for a description of the other problems found). According to Gangemi et al., the solution is a "sweetening" of WordNet accomplished by the use of OntoClean [Guarino and Welty, 2004], a methodology for validating the ontological adequacy of taxonomic relationships. They conclude that it is possible to do some "cleaning" of the errors found in WordNet. However they cannot say whether a "clean" reconstruction will have a positive impact on the performance for the applications using it.

3.6 Summary and Discussion

This chapter began with a discussion of the notion of ontologies, narrowing the definition to a philosophical sense and a knowledge engineering sense. The brief introduction to the philosophical notion of ontology stated the foundation while the computer science notion was defined as a pragmatic view with an emphasis on usability rather than philosophy.

A variety of formalisms for the representation of ontologies have been presented and discussed. The differences between these approaches are not huge, but rather dependent upon the "spirit" of the formalism. One could argue that most of the formalisms could be stretched to subsume one another, but it has also been argued that it would be better to choose the formalism with the correct "spirit" right from the beginning, rather than having to do all of the bending and stretching subsequently.

For the purpose of this thesis, we have chosen the lattice-algebraic description language as it fits perfectly into the notion of generative ontologies. The "spirit" of the description language Ontolog supports the generative aspect of the ontologies used here and can be mapped directly into the ontology. The simple notation, which is similar to feature structures, is easy to understand even when it is completely detached from its framework, and is therefore also useful for presentation. Furthermore, the lattice-algebraic notation supports meet and join intrinsically (the greatest lower bound and least upper bound, respectively), where especially the former has a central role in measuring similarity derived from the knowledge about relations between concepts in ontologies.

There is a close connection between the logico-algebraic framework and descriptions logic, where the latter has become the standard in the Semantic Web area, with the OWL Web Ontology Language [Bechhofer *et al.*, 2004] as its newest branch. If solely description logic reasoning capabilities are used in the retrieval process, we restrict ourselves to an approach in which querying is restricted to model checking, thus removing the possibility of partial fulfillment of the query and thereby also the means for graded ranking. This, of course, would also be the case if we wanted to do reasoning using the lattice-algebraic representation. Instead, the aim is to transform the representation into directed graphs in which well-known algorithms can be used to compute, for instance, the shortest path between nodes. Naturally, any of the representations presented can be transformed into directed graphs, but the graphical nature of the lattice-algebraic representation makes this very easy compared to, for example, description logic.

One of the major improvements of description logic is the emphasis on decidability and the possibility for implementing fast model checkers (see e.g. [Horrocks, 1998]). With regard to ontology, modeling such tools is preferable for checking consistency. Obviously, consistency checks could naturally also be done in the logico-algebraic framework, but without a set of freely available tools. One overall solution could then be to use the logico-algebraic framework for representing the ontologies, the description logic framework for modeling the ontologies, and directed graphs for measuring similarity.

Chapter 4

Descriptions

The indexing process was only briefly touched upon in Chapter 2 and will be treated in more detail here. An information retrieval system presupposes indexing and the system's performance depends on the quality of this indexing. The two main challenges in indexing are 1) to create representative internal descriptions of documents in the provided description notation/language and 2) to organize these descriptions for fast retrieval.

In this chapter we will mainly focus on the first challenge and discuss different types of notations and generation techniques. The aim is to define a notation useful for representing the descriptions needed in ontology-based information retrieval and to make an outline of the generation techniques.

A *description* of a document is comprised of information found by analyzing (indexing) the document. The constituents in a description, the *descriptors*, are collected in a structure in accordance with the *description notation.* A simple example of the latter is a "bag of words" in which a description of a document is a collection of words (descriptors) appearing in the document.

Descriptions of documents in information retrieval are supposed to reflect the documents' content and establish the foundation for the retrieval of information when requested by users. Completeness with respect to the description of content is obviously very difficult to achieve, since only the authors know the true content of their writings. Descriptions are therefore always a type of surrogates in relation to the original documents. We define fidelity of descriptions (surrogates) as the description's ability to represent the content of the document it refers to. *Perfect fidelity* would then be the case where the description represents the content of a document exactly, which is generally impossible, both in practice and in principle, since the only completely accurate representation of a document's content is the content itself. All other representations are inaccurate; they inevitably contain simplifying assumptions [Davis *et al.*, 1993]. .

A common way of defining the properties of a description notation is by using the comparative terms *exhaustivity* and *specificity*, where *exhaustivity* is a property of index descriptions, and *specificity* is a property of descriptors [Jones, 1972].

Exhaustivity refers to the degree to which we recognize the different concepts in documents [Cleverdon and Mills, 1997]. Thus, the degree of *exhaustivity* for a document description increases if, for instance, more descriptors are assigned to the description, provided that these descriptors add new concepts to the description. The maximum degree of *exhaustivity* would then be the case where all concepts are recognized and assigned to the description; hence increasing the degree of exhaustivity would also increase the fidelity of the description. Normally, an optimum degree of *exhaustivity* for a given document collection is defined as the average number of descriptors per document where the likelihood of requests matching relevant documents is maximized and too many false matches are avoided. This is a conceptual definition of *exhaustivity*, which is obviously very interesting in the context of an ontological retrieval framework. However, most retrieval frameworks are word-based (lexical) and cannot refer to this conceptual definition; hence, a statistical counterpart is used and the *exhaustivity* of a document's description is the number of terms it contains [Jones, 1972]. In this redefinition of *exhaustivity* the maximum degree of *exhaustivity* would not be necessary, as it is with the conceptual definition, to bring the description closer to perfect fidelity, since lexical terms contain the possibility of ambiguity with respect to meaning.

Specificity of a descriptor is defined as the degree to which it reflects the precise generic level of the concept it stands for [Cleverdon and Mills, 1997]. For instance, the descriptor "*private school*" would be considered more specific than the broader descriptor "*school*" and would therefore refer to a smaller collection of documents, since the descriptor "*school*" includes (subsume) the descriptor "*private school*". Specificity thus refers to the ability of descriptors to discriminate one document from another (defined as *resolving power* in Section 2.1.1). Increasing the degree of specificity of descriptors in a description would also increase the fidelity of the description. Again, this is a conceptual definition, and like *exhaustivity* it has a statistical counterpart in the term *specificity*, which is the number of documents to which it pertains [Jones, 1972].

Finally, a measure is needed which refers to the content of the description itself. The use of either the conceptual or the statistical definition of exhaustivity and specificity would make a significant difference in the representation, since it would be either a semantic or a lexical description. The descriptors of a description can refer to single words, multi-words, or both, which again would define descriptions with different contents. Obviously, the different definitions of the content of descriptions are countless. Inspired by the notion of high and low levels of programming languages, the lowest level can be defined as the string that contains the document. Any alteration of the lowest level would therefore lead to a higher level. Note that a transformation from one level to a higher level does not necessary lead to a higher degree of fidelity and vice versa.

4.1 Indexing

The process of assigning descriptions to documents in an information retrieval system is called *indexing*.

In Figure 2.2 the indexing process is divided into two parts, the conceptual analysis and a transformation process, where the former extracts information from documents and the latter creates descriptions in accordance with the description notation.

Indexing can be performed either manually or automatically and in either a so-called controlled or an uncontrolled manner. In manual indexing, experts assign the descriptions, while automatic indexing is performed by computers. Uncontrolled indexing refers to an indexing process with no limitations on the form and the content of the descriptors, while controlled indexing restricts the description of documents to some predefined amount of information, e.g. a set of keywords, a thesaurus, etc.

In addition to the controlled indexing by a predefined set of descriptors, manual indexing often uses a set of rules to guarantee homogeneous indexing. An example of such a rule is that whenever two similar descriptors apply, the most specific one must be used. Rules like this are especially bound to traditional library systems where descriptions, on average, have few descriptors.

The automated indexing process is normally not controlled, at least not in the same sense as in manual indexing, nor is it rarely completely uncontrolled. Different kinds of methodologies are often used to reject information without significance for the content of a given document.

A mix of the two is sometimes called semi-automated indexing, which is often done by automated indexing that is subsequently analyzed manually. The purpose of this type of indexing is to reduce the human effort when indexing larger amounts of data, while still maintaining expert validation. Using (semi-)automated indexing is further justified when one of the main problems with manual indexing is taken into consideration: the lack of scalability to huge collections of information, for instance, to the Internet due to the enormous amount of human resources that would be needed. Another problem with manual indexing is the presuppositions of the experts and their personal biases, which tend to emphasize some subjects and suppress others [Edmundson and Wyllys, 1961]. The major advantage of manual indexing is its high quality and homogeneity. Moreover, manual indexing is a realistic approach for smaller document collections, especially when performed by experts that not only have insight into the domain but also have skills in providing representative descriptions. In this context, the experts are typically librarians.

The main problem with automated indexing is obtaining high quality descriptions that are representative and have a minimum of errors, especially indications that contradict with the actual content of the documents. The major advantage of automated indexing is the low cost of human resources. Furthermore, experiments have shown that on large full-text document resources, automated indexing is superior to the manual indexing, even with respect to retrieval evaluation [Salton and Buckley, 1997]. Because only automated indexing is taken into consideration in this thesis, the term *indexing* refers to the above whenever it is used.

The foundation of many conventional retrieval systems is some alteration of a very simple representation of documents in which descriptions are a collection of the words appearing in the documents. A number of devices intended to improve recall or precision can be added to this basic structure. Cleverdon and Mills [1997] present a list of thirteen devices which, when introduced to an uncontrolled vocabulary of simple terms, tend to broaden the class definition (numbers 1-10) and thus increase recall, or narrow the class definition (numbers 11-13) and thus increase precision:

1. Confounding of true synonyms.

2. Confounding of near synonyms; usually terms in the same hierarchy.

3. Confounding of different word forms; usually terms from different categories.

4. Fixed vocabulary; usually takes the form of generic terms, but may use "metonymy", for example, representing a number of attributes by the thing possessing them.

5. Generic terms.

6. Drawing terms from categories and, within these, facets; this controls the generic level of terms, and to a certain degree controls synonyms.

7. Representing terms by analytical definitions (semantic factors) in which inter-relations are conveyed by relational affixes or modulants; the generic level will usually be more specific than when control is done by categories.

8. Hierarchical linkage of generic and specific terms and, possibly, of coordinate terms.

9. Multiple hierarchical linkage, i.e. linking each term to a number of different generic heads.

10. Bibliographical coupling and citation indexes; these are also ancillary devices which indicate the existence of wider classes, the latter reflecting the usage of documents and a probability of relevance arising from this.

11. Correlation of terms: although implicit in some form in all practical indexing, this is not inevitable, i.e. the use of a single term to define a class may retrieve quickly and economically if the term is sufficiently rare in the context of the system.

12. Weighting, i.e. attempts to express the particular relevance of each concept used in indexing a document in relation to the whole document (see 2.1.1).

13. Indicating connections between terms (interlocking).

Some of the devices in the above list are well-known and used frequently in many retrieval systems. The aim of this thesis is to introduce external knowledge, especially ontologies, in information retrieval which makes some of the devices listed particularly interesting.

Devices 8 and 9 refer to the use of taxonomies as a way to broaden the descriptions. Device 13 refers to connections between terms and is defined as a narrowing device that increases specificity as it introduces more compound terms, as in the example with "*school*" and "*private school*" from the definition of specificity, where the connection between "*private*" and "*school*" are interpreted as the semantic relation "*characterized by*", i.e. *school*[CHR:*private*].

4.2 Ontology-based Indexing

One way of introducing external knowledge into information retrieval is by using fixed vocabularies in controlled indexing (device 4), for instance, by means of a list of keywords that reflect knowledge about the domain. A further step in this direction is to apply taxonomies, i.e. hierarchically structured, rather than simple collections of keywords (devices 8 and 9). Ontologies can be considered as a kind of extended taxonomies that also are comprised of complex concepts, which typically correspond to compound or modified key words, e.g. *school*[CHR:*private*] (device 13). Thus, an ontological base means that it is possible to obtain an even richer approach to knowledge-based indexing. However, ontology-based indexing also involves new challenges in connection with indexing, for example, the inclusion of some kind of semantic analysis, which is necessary if compound concepts are to be identified.

The description notation also needs to be modified to accommodate for conceptual expressions. The alteration of the descriptions is first and foremost from a lexical to a semantic representation and thus in a certain sense introduces high-level descriptions. This also means an increased degree of fidelity, since the degree of specificity is increased due to the disambiguation of meanings and the union of concepts into compound concepts.

The two main challenges in utilizing ontologies in information retrieval are 1) to map the information in documents and queries into the ontologies and 2) to improve retrieval by using knowledge about relations between concepts in the ontologies. The remainder of this chapter will focus on solving the former, while the latter is the topic of the next chapter.

The semantic analysis needed should somehow recognize concepts in the documents and then map them into the ontologies, and by so doing reveal the precise meaning, which is called *word sense disambiguation*. Both of these tasks are well-known parts of natural language processing.

Natural language processing, which is a subfield of artificial intelligence and linguistics, studies the problems inherent in the processing of natural language devoted to making computers "understand" statements written in human languages. The tasks in natural language processing can be grouped in many ways, for example, using Jurafsky and Martin's [2000] grouping, which consists of four parts: *words*, *syntax*, *semantics*, and *pragmatics*. Words are the building blocks of natural language; syntax is the study of formal relationship between words; semantics is the study of the meaning of linguistic utterances; and pragmatics is the study of the relation between language and context-of-use.

In natural language processing, there are generally two main approaches, deep approaches and shallow approaches. Deep approaches presume access to a comprehensive body of world knowledge. These approaches are not very successful in practice, mainly because access to such a body of knowledge does not exist, except in very limited domains. Shallow approaches, which do not try to understand the text, only consider the surroundings in the text being analyzed. One guiding principle in this regard is to use "simple" rules to resolve knowledge. These rules can either be automatically derived by using sense-tagged training corpuses or be manually defined by experts. This approach, while theoretically not as powerful as deep approaches, gives superior results in practice due to the limited requirements of world knowledge.

The aim is to reveal partial semantic information in the documents and then map it into the ontologies. The hypothesis is that even minor steps towards revealing the semantics can be useful in information retrieval. Achievements are measured by the retrieval improvements gained – not by how close we can get to a complete semantic analysis. In information retrieval the main goal is to retrieve information that satisfies the users information need, thus any system enhancements that improve the retrieval for some subset of the queries posed to the system can be considered as a refinement of the system in general, provided that such improvements do not have any negative influence on results on another subset of queries.

A simple semantic analysis would be to recognize specific word classes, e.g. nouns, verbs, adjectives, etc. and then map them into the ontology. However, disambiguation of word classes does not necessarily imply disambiguation of sense. Hence, some methods used to resolve unambiguous sense are also required in order to map to a unique position in the ontology. An example of a more complex analysis would be the recognition of connected knowledge, e.g. noun phrases, verb phrases, adjective phrases, whole sentences, etc. To obtain this analysis, methods are needed that are designed to reveal the syntax of interrelations between concepts in connection with sense disambiguation.

The natural language processing tasks in use in the indexing described here are morphological analysis, part-of-speech (word class) tagging, syntactical parsing with context-free grammars and word sense disambiguation. While tasks are only touched upon lightly in this thesis, different specific approaches are suggested for solving the tasks used without an in-depth analysis of the underlying techniques, much less the theory behind. For a good introduction to the details of natural language processing see, e.g. Jurafsky and Martin [2000].

4.2.1 The OntoQuery Approach

In the OntoQuery project [Andreasen *et al.*, 2000; Andreasen *et al.*, 2002; OntoQuery, 2005], queries take the form of natural language expressions and the system is intended to retrieve text segments whose semantic content matches the content of the noun phrases in the query. In OntoQuery sentences are the basis of the indexing of documents. Queries, likewise, are assumed to be sentences where the query evaluation compares sentences on the basis of their noun phrases.

The aim of the linguistic and conceptual analysis of a text is to identify simple as well as complex concepts corresponding to the noun phrases occurring in the text and to ensure that noun phrases with (nearly) identical conceptual content with respect to the ontology are described by the same description. This goal, which clearly goes beyond synonym recognition and identification of morphological variants, calls for a more comprehensive linguistic and ontological analysis of the text material [Andreasen *et al.*, 2004]. The emphasis on noun phrases is motivated by the fact that these have clear conceptual content that can be captured in an ontology-based framework.

The ideas that led to the first OntoQuery prototype were for a text fragment (e.g. a sentence) that a simple form description could express the content by means of a nested set of words from the sentence:

$$D = \{D_1, \ldots, D_n\} = \{\{D_{11}, \ldots, D_{1m}\}, \ldots, \{D_{n1}, D_{n2}, \ldots, D_{nm}\}\}$$

where each descriptor D_i is a set of concepts $D_{i1}, \ldots, D_{im}$.

To generate descriptions, text fragments are prepared by a parser that employs a knowledge base. This parser can in principle be on a scale from a simple word recognizer to a complex natural language parser that maps the full meaning content of the sentence into an internal representation. Since the issue in the OntoQuery project prototype is information retrieval, the idea is, of course, to grab fragments of content rather than represent full meaning, and that the building stones are concepts. The level of descriptions should be understood considering this aim.

Parsing involves a part-of-speech tagger, a noun-phrase recognizer and a subcomponent that builds descriptions in the description language, as illustrated in Figure 4.1. Tagging is performed by an implementation of Eric Brill's tagger [Brill, 1995]; the noun phrase recognition is performed by the chunk parser "Cass" [Abney, 1996]; while the noun phrase grammar has been developed manually on the basis of the occurrence of various noun phrase types in the PAROLE corpus [Ruimy *et al.*, 1998] and covers noun-phrase chunks extending from the beginning of the constituent to its head and includes post modifying prepositional phrases.

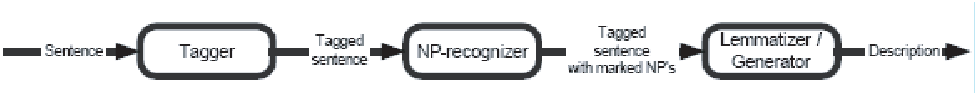


Figure 4.1: *The process of generating descriptions in the* OntoQuery *prototype*

Descriptions are generated from the result of the noun-phrase recognizer through morphological processing, part-of-speech filtering and grouping into descriptions. The grouping simply corresponds to the noun phrases recognized; thus only words belonging to noun phrases are included in the descriptions.

The following phrase is an example:

Physical well-being caused by a balanced diet

A description consisting of nouns and adjectives in a simple form without taking into account that the framing of noun phrases in the sentence could be:

{"*physical*", "*well-being*", "*balanced*", "*diet*"}

With the framing of noun phrases the descriptors can be gathered to lead to the nested set description:

{{"*physical*", "*well-being*"}, {"*balanced*", "*diet*"}}

.

The set of words representing a single noun phrase can be seen as an approximation where the relations between concepts are defined mainly by what prepositions are left unspecified. A set of words is thus considered as an abstraction of a concept due to the way it is applied in the prototype. This implies that in addition to concept inclusion via the ISA relation of the ontology, a concept inclusion also exists as derived from set inclusion.

In the next and current version of the OntoQuery prototype, the level of the descriptions in use was changed into a collection of Ontolog expressions. In order to reveal noun phrases as compound Ontolog expressions the parsing should in principle introduce recognition of interrelations between concepts. Since the methodologies for this purpose were not available for the project, a simple shortcut was chosen. The simplified principle consisted of two-phase processing, with the first phase basically consisting of a noun phrase bracketing, and the second, of an individual extract of concepts from the noun phrases. A naive but useful second phase was to extract nouns and adjectives only and combine them into *noun* CHR *adjective* pattern concepts (CHR representing a "characterized by" relation). Thus, the above nested set representation would be transformed into:

$$\{\textit{well-being}[\text{CHR}:\textit{physical}], \textit{diet}[\text{CHR}:\textit{balanced}]\}$$

Obviously, the method is an oversimplification and generates erroneous compound concepts, e.g. "*criminal lawyer*" would be transformed into:

$$\textit{lawyer}[\text{CHR}:\textit{criminal}]$$

which is rarely the right interpretation. However, this was not crucial for the testing of query evaluation with respect to this new kind of descriptions.

One issue which is not handled properly in the OntoQuery prototype is word sense disambiguation. In order to map the revealed knowledge into the ontology, a mapping between the concepts in the ontology and a dictionary, which serves as the vocabulary of the prototype, is used. This mapping, however, disambiguates by only picking the first sense (the one with the lowest identifier) if more than one is present for a particular word. The reason why this method is useful at all is that the nutrition domain in focus in OntoQuery does not have many ambiguities; hence the ambiguity problem does not dominate and significant testing examples can easily be found.

4.2.2 Word Sense Disambiguation

Word sense disambiguation involves the association of a given word in a text with a definition or sense. The problem of word sense disambiguation has been described as *AI-complete*, that is, a problem which can be solved only by first resolving all the difficult problems in artificial intelligence, such as the representation of common sense and encyclopedic knowledge [Nancy and Véronis, 1998]. Obviously, this is not the target here. Rather, the aim of this section is to give some examples of methodologies for mapping concepts into the ontology.

We have already described two simple methodologies for disambiguating word senses. In the previous section the simple approach used in the current OntoQuery prototype where senses are chosen at random was looked at. Earlier, in Section 3.5.3 we described how the use of sense frequencies increased the quality of disambiguation from 45% to 69% correct senses compared to purely guessing.

A completely different approach to word sense disambiguation is to apply the meaning structure of language, for example, the predicate-argument structure, which is useful in disambiguation. Take, for instance, the verb "*eat*". The use of this verb in texts can be seen as a predicate describing *Eating* with one or more arguments. The arity of a predicate can vary dependent upon the use, for instance, "I ate" or "I ate a sandwich", where the number of arguments are one and two respectively. Events can be represented in predicate calculus and the *Eating* with two arguments might look as follows:

$$\exists e, x, y(Eating(e) \wedge Agent(e, x) \wedge Theme(e, y))$$

where e is an event – in this case the *Eating* event, and x and y are the two arguments. In this example, the arguments are expressed by the *thematic roles*; *Agent* and *Theme*, defined as the volitional causer of an event and the participant most directly affected by an event, respectively. These semantic roles can be restricted by semantic constraints, denoted *selectional restrictions*. An obvious restriction on the theme role for the *Eating* event would be restricting edible things:

$$\exists e, x, y(Eating(e) \wedge Agent(e, x) \wedge Theme(e, y)) \rightarrow y \text{ ISA } EdibleThing$$

Consider the example of the WordNet synset {"food", "nutrient"} with the following gloss "any substance that can be metabolized by an organism to give energy and build tissue". This synset could serve as the top-most concept acceptable as second arguments for the *Eating* event. The selectional restriction will in this case reject senses not subsumed by the {"food", "nutrient"} synset, hence ruling out non-edible senses for the second argument to the *Eating* event.

Selectional restrictions can be defined at any level, spanning from very general to very specific restrictions in either general or specific domains, as well as be associated to hierarchies, such as taxonomies and ontologies. The advantage of combining selectional restrictions with hierarchies is that the restrictions are applied not only to the given concept, but also to all subsumed concepts, as indicated in the above example. Furthermore, one benefit of applying selectional restrictions is that they can be used for rejecting erroneous senses and thereby for reducing ambiguity, which is otherwise difficult to recognize. The major drawback of selectional restrictions is that natural language is difficult to restrict, e.g. "*...you can't eat gold for lunch...*", which is a perfectly well-formed phrase, but "*eat gold*" would clearly violate the selectional restriction on "*eat*"as defined above. Another problem is that selectional restrictions have many requirements in order to be useful in large-scale practical applications. Even with the use of WordNet, the requirements are unlikely to be met for complete selectional restriction information for all predicate roles and for complete type information for the senses of all possible fillers.

In large-scale applications, selectional restrictions are therefore primarily useful in combination with other approaches. One such approach is machine learning as it is a commonly used method in word sense disambiguation [Nancy and Véronis, 1998]. With machine learning, problems can be solved using either supervised or unsupervised learning, where supervised learning refers to learning from predefined training data, while unsupervised learning does not presuppose external knowledge. Only the basic idea of word sense disambiguation based on supervised learning is outlined here. In most machine learning approaches, the initial input consists of the word to be disambiguated, the *target word*, along with a portion of the text in which it is embedded, the *context*. The input is normally part-of-speech tagged and lemmatized, and a specific amount of the text surrounding the target word (the context) is selected, a so-called window, (often) with the target word in the center.
The following text fragment can be given as an example:

An electric guitar and bass player stand off to one side, ...

where the input to the leaning process for the target word "*bass*", consisting of the two words to the right and left, would be

(guitar/NN, and/CC, player/NN, stand/VB)

where NN, CC, and VB are part-of-speech tags for nouns, coordinating conjunctions, and verbs, respectively[1].

This can easily be transformed into a simple *feature vector* consisting of either numerical or nominal values, which would be appropriate for use in most learning algorithms. The naïve Bayes' classifier is one such commonly used learning approach for word sense disambiguation. The premise is that choosing the best sense for an input vector amounts to choosing the most probable sense given that vector, formally:

$$\hat{s} = \text{argmax}_{s \in S} P(s|V) \tag{4.1}$$

[1] Refers to the part-of-speech tags in the Penn Treebank Tag Set.

where S is the set of appropriate senses for the target word, V is the input feature vector, and $\hat{s}$ is the best (most probable) sense. By using the Bayesian rule and ignoring factors that are constant for all senses we then get the following for a feature vector of n elements:

$$\hat{s} = \text{argmax}_{s \in S} P(s) \prod_{j=1}^{n} P(v_j|s) \tag{4.2}$$

where $P(s)$ is the priori probability of sense s and v_j is an element in the feature vector. $P(s)$ can be obtained from the proportion of each sense in the sense-tagged training corpus.

In the "*bass player*" example, the individual statistics needed might include the probability of the word "*player*" occurring immediately to the right of a use of one of the "*bass*" senses, or the probability of the word "*guitar*" one place to the left of the use of one of the senses for "*bass*". This is, of course, only one example of word sense disambiguating using supervised learning. For a survey of the variety of approaches, see e.g. Nancy and Véronis [1998].

Finally, a range of disambiguation methods that draw on lexical resources has been developed. In Voorhees [1998], the notion of *hood* introduced by George Miller is used to determine the most likely sense of a given word. A *hood* is an area in WordNet in which a word is unambiguous. More precisely, to define the hood of a given synset s, consider the set of synsets and the hyponymy relation in WordNet as vertices and directed edges in a graph. The hood of s is then the largest connected sub-graph that 1) contains s, 2) contains only descendants of an ancestor of s, and 3) contains no synset that has a descendant that includes another instance of a member of s as a member [Voorhees, 1998].

For example, consider the piece of WordNet shown in Figure 4.2 where the gray boxes denote the synsets with the word "*board*" as a member. The hood of the synset for the "*committee*" sense of "*board*" is rooted at the synset {*group*}, and thus the hood for that sense is the entire hierarchy (all ancestors) in which it occurs. The hood of the "*computer circuit*" sense of "*board*" is rooted in the ancestor {*circuit, electrical circuit, electric circuit*}, and the root of the "*control panel*" sense of "*board*" is rooted at the synset itself, etc. Some synsets may have more that one hood if they have more than one parent, while they have no hood if the same word is a member of both the synset and one of its descendants. A simple way of disambiguating words is to find the hood where most of the surrounding context also appears (see Voorhees [1998] for more details).

Another approach, similar to the use of *hoods*, is to apply a measure of distance between concepts in an ontology (the topic of the next chapter). Given such a measure of similarity between concepts, the sense closest to the

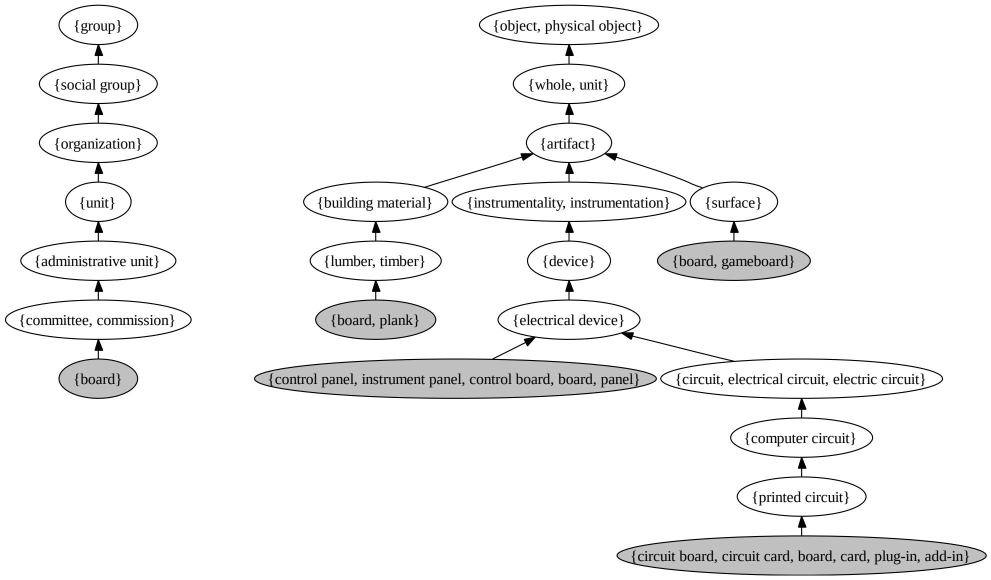


Figure 4.2: *The WordNet hierarchy for five different senses of "board"*

context of a target word can be selected:

$$\hat{s} = \text{argmin}_{s \in S} \left(\sum_{i=1}^{n} dist(s, c_i) \right) \tag{4.3}$$

where $dist(x, y)$ is some kind of concept-concept similarity measure, S is the set of appropriate senses for the target word, c_i a concept in the window, n the number of concepts in the window, and $\hat{s}$ the sense closest to the context of the window (see e.g. [Agirre and Rigau, 1996; Miller *et al.*, 1994; Resnik, 1995]).

In cases where the disambiguating methods cannot determine one sense, the method must either abandon the attempt or introduce rules for selection between the different similar possibilities. One solution could be *boosting*, the idea of combining several (moderately accurate) methods into a single highly accurate approach. A combination of, for instance, sense frequencies, selectional restrictions, machine learning approaches, and ontological approaches could lead to a more accurate methodology, especially in the case where one particular sense could not be pointed out (see e.g. [Miller *et al.*, 1994; Escudero *et al.*, 2000; Nancy and Véronis, 1998]).

In this thesis the word sense disambiguation determines the correct mapping of words in texts to senses in the ontology. One major problem in doing this concerns the granularity (level of detail) of senses in the ontology. WordNet is very fine-grained and some of the senses are too similar, almost

Word	Synset	Gloss	Hypernym	Hypernym Gloss
bass	{*bass*, *basso*}	an adult male singer with the lowest voice	{*singer*, *vocalist*, *vocalizer*, *vocaliser*}	a person who sings
bass	{*bass*, *bass voice*, *basso*}	the lowest adult male singing voice	{*singing voice*}	the musical quality of the voice while singing

Table 4.1: *Two of the senses associated with the word "bass" in WordNet and their hypernyms*

indistinguishable. For example, this is the case with two of the senses of the noun "*bass*" in WordNet as shown in Table 4.1, where the disambiguation has to distinguish between the voice of a singer and the voice itself.

In practice, it is not always possible for automatic disambiguation to distinguish between such fine-grained senses, and even humans have problems making the right choice in some cases. This is a problem we encountered several times when trying to judge whether the automated disambiguation had made the right choice. One solution to this granularity problem would be to automatically cluster WordNet senses according to their similarity (see e.g. [Mihalcea and Moldovan, 2001; Tomuro, 2001; Agirre and de Lacalle, 2003]). For this purpose, a measure of similarity between concepts in the ontology is needed, a topic that will be looked at in the next chapter.

4.2.3 Identifying Relational Connections

The introduction of compound concepts requires, in addition to syntactical knowledge, a means for identifying the semantic relations that tie the concepts together in order to transform word patterns into Ontolog expressions. The phrase "Physical well-being caused by a balanced diet", for example, can be transformed into a single Ontolog expression:

well-being[CHR:*physical*][CBY:*diet*[CHR:*balanced*]]

if we are able to identify the semantic relations between noun phrases and between constituents inside the noun phrase.

The fragment from an ontology in Figure 4.3 shows a visualization of the above compound concept. This figure also shows how compound concepts are decomposed and thus how they can be merged into the ontology. This topic is looked at in the next section.

There is a significant difference between recognizing relations connecting noun phrases and relations connecting constituents inside noun phrases. The former is a very complex challenge as the relations between noun phrases should reflect all possible relations bound to the verbs of natural language.

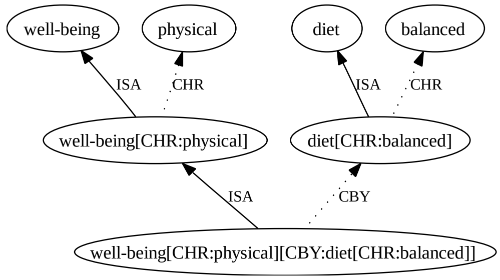


Figure 4.3: *An ontology fragment showing the concept well-being[*CHR*:physical] [*CBY*:diet[*CHR*:balanced]]*

Besides a description of a promising machine learning approach in Section 8.5, this problem will not be discussed further in this thesis.

The internal relations in noun phrases can be subdivided into pre-modification and post-modification, which roughly reflects adjective phrases before the head and prepositional phrases after, respectively.

WordNet divides adjectives into two main groups, descriptive and relational. The descriptive adjectives are the common adjectives, e.g. "*small*", "*beautiful*", "*possible*", etc., and they characterize the noun they modify, e.g. "*small picture*", "*beautiful flower*", etc. The obvious semantic relation would therefore be CHR ("*characterized by*"). Alonge et al. [2000] define the relation for relational adjectives as a PERTAINS_TO relation, e.g. "*musical instrument*" and "*dental hygiene*", which can be interpreted as a WRT ("*with respect to*") relation. This would, for instance, change the erroneous interpretation "*lawyer*[CHR:*criminal*]" of "*criminal lawyer*" into the more correct interpretation "*lawyer*[WRT:*criminal*]".

Quantifiers, e.g. "all", "some", "few", etc, are one minor group of descriptive adjectives that can definitely cause problems in this rough division. The relation between quantifiers and the noun they modify cannot be interpreted as a characterization of the CHR relation, e.g. "$professor$[CHR:few]" is an erroneous interpretation of the phrase "*few professors agreed*". The solution to this problem would be to exclude the set of quantifiers from the rule that interprets descriptive adjective as the CHR relation, which is easily done in WordNet since quantifiers are marked. Another argument supporting this is that the set of quantifiers are classified by some linguists as determiners rather than as adjectives because, for instance, they appear syntactically in pre-adjectival positions similar to that of determiners.

Identifying the semantic relations expressed by prepositions is much more complicated than for adjectives. Simple one-word prepositions in English constitute a relatively small class of words and it should therefore be possible to define rules concerning the use of the most frequently used simple prepositions.

According to [Quirk *et al.*, 1985a], the prepositional meanings can roughly be divided into the following main semantic categories:

- Dimension
- Time
- Cause/Purpose
- Means/Agentive

The first two items, dimension and time, are the most interesting in relation to simple prepositions in noun phrases. The main problem is that the same prepositions are used in different categories, e.g. “the car on the road” and “the meeting on Monday”, where “on” falls into the dimension and time categories respectively. One way to differentiate between the two is by using an ontology. A simple ontology-based disambiguation rule could then be: If the relation involves temporal concepts, like “Monday”, the relation for the preposition “on” is TMP (“*temporal*”) relation, e.g. “*meeting*[TMP:*Monday*]”, otherwise the relation is LOC (“*location*”), e.g. “*car*[LOC:*road*]”.

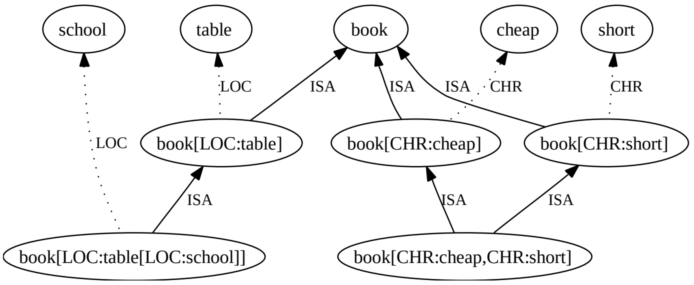


Figure 4.4: *An ontology fragment visualizing the interpretation of nested, book*[LOC*:table*[LOC*:school*]]*, and listed, book*[CHR*:cheap,* CHR*:short*]*, concepts*

Prepositional relations are generated as nested ONTOLOG expressions whenever the same kind of relation is used twice, e.g.“the book on the table in the school” is interpreted as the expression “*book*[LOC:*table*[LOC:*school*]]”, while simple pre-modifications are interpreted as lists, and “the short and cheap book” is interpreted as the expression “*book*[CHR:*cheap*, CHR:*short*]”. The different interpretations of nested and listed ONTOLOG expressions are shown in Figure 4.4.

The simple method used in the ONTOQUERY prototype where the CHR is used for all pre-modifications and the LOC for all post-modifications can be improved significantly by the observations described in this section.

4.3 Instantiated Ontologies

The main objective in the modeling of domain knowledge is for the domain expert or knowledge engineer to identify significant concepts in the domain.

Ontology modeling in the present context - the work presented here as well as the OntoQuery project - is not a key issue. However, in relation to this issue we present the notion of Instantiated Ontology as a (naive) substitution for, or a supplement to, expert modeling[2]. The modeling consists of two parts. First, an inclusion of knowledge from available knowledge sources into a general ontology and second, a restriction to a domain-specific part of the general ontology. The first part involves the modeling of concepts in a generative ontology using different knowledge sources. In the second part, a domain-specific ontology is retrieved as a sub-ontology of the general ontology. The restrictions on this sub-ontology are built based on the set of concepts that appear (are instantiated) in the document collection and the result is called an instantiated ontology.

4.3.1 The General Ontology

Sources for knowledge base ontologies may have various forms. Typically, a taxonomy can be supplemented with, for instance, word and term lists as well as with dictionaries for the definition of vocabularies and for handling the morphology.

Without going into detail on the modeling here, we assume the presence of a taxonomy in the form of a simple taxonomic concept inclusion relation ISA_{KB} over the set of atomic concepts $\mathbf{A}$. ISA_{KB} and $\mathbf{A}$ express the domain and world knowledge provided. ISA_{KB} is assumed to be explicitly specified, e.g. by domain experts, and would most typically not be closed transitively.

Based on $\widehat{\text{ISA}}_{\text{KB}}$, the transitive closure of ISA_{KB}, a relation can be generalized concerning all well-formed terms of the language $\mathbf{L}$ by the following:

- if $x \ \widehat{\text{ISA}}_{\text{KB}} \ y$ then $x \leq y$
- if $x[\ldots] \leq y[\ldots]$ then also

  $$x[\ldots, r\colon z] \leq y[\ldots], \text{ and}$$
  $$x[\ldots, r\colon z] \leq y[\ldots, r\colon z],$$

- if $x \leq y$ then also

[2] The presentation of Instantiated Ontologies in Section 4.3 is a slightly modified rendering of the original presentation in [Andreasen *et al.*, 2005a]. The notion of Instantiated Ontologies is partly based on and stimulated from the notion of Similarity Graphs introduced in [Andreasen *et al.*, 2003c].

$$z[\dots, r\colon x] \leq z[\dots, r\colon y]$$

where repeated ... in each case denotes zero or more attributes of the form $r_i\colon w_i$.

The general ontology $O = (\mathbf{L}, \leq, \mathbf{R})$ thus encompasses a set of well-formed expressions $\mathbf{L}$ derived from the concept language with a set of atomic concepts $\mathbf{A}$, an inclusion relation generalized from an expert provided relation ISA_{KB} and a supplementary set of semantic relations $\mathbf{R}$, where for $r \in \mathbf{R}$ it is obvious that $x[r\colon y] \leq x$ and $x[r\colon y]$ are in relation r to y. Observe that $\mathbf{L}$ is infinite and that O is thus generative.

4.3.2 The Domain-specific Ontology

Apart from the general ontology O, the target document collection contributes to the construction of the domain ontology. We assume a processing of the target document collection, where an indexing of text in documents, formed by sets of concepts from $\mathbf{L}$, is attached. In broad terms, the domain ontology is a restriction of the general ontology to the concepts appearing in the target document collection. More specifically, the generative ontology is, by means of concept occurrence analysis over the document collection, transformed into a domain specific ontology restricted to include only the concepts instantiated in the documents covering that particular domain.

This reduction has the obvious advantage of reducing the number of concepts we have to consider when deriving similarity for use in topic-based surveying and content-based querying of the document collection. The intuition is that concepts in the knowledge-based ontology that are not present in the domain do not contribute to similarity between concepts present in the object document collection. Since these concepts are not used for the description of the semantics of the objects, queries using these concepts have potentially empty answers. Thus the domain specific ontology is introduced as an "instantiated ontology" of the general ontology with respect to the target document collection.

The instantiated ontology $O_{\widehat{I}}$ appears from the set of all instantiated concepts I, first by expanding I to $\widehat{I}$ - the transitive closure of the set of terms and sub-terms of term in I - and second by producing a sub-ontology consisting of $\widehat{I}$ connected by relations from O between elements of $\widehat{I}$.

The sub-terms of a term c are obtained by the decomposition $\tau(c)$. $\tau(c)$ is defined as the set of all sub-terms of c, which thus includes c and all attributes of subsuming concepts for c.

$$\tau(c) = \{c\} \cup \{x | c \leq x[\dots, r\colon y] \vee c \leq y[\dots, r\colon x], x \in \mathbf{L}, y \in \mathbf{L}, r \in \mathbf{R}\}$$

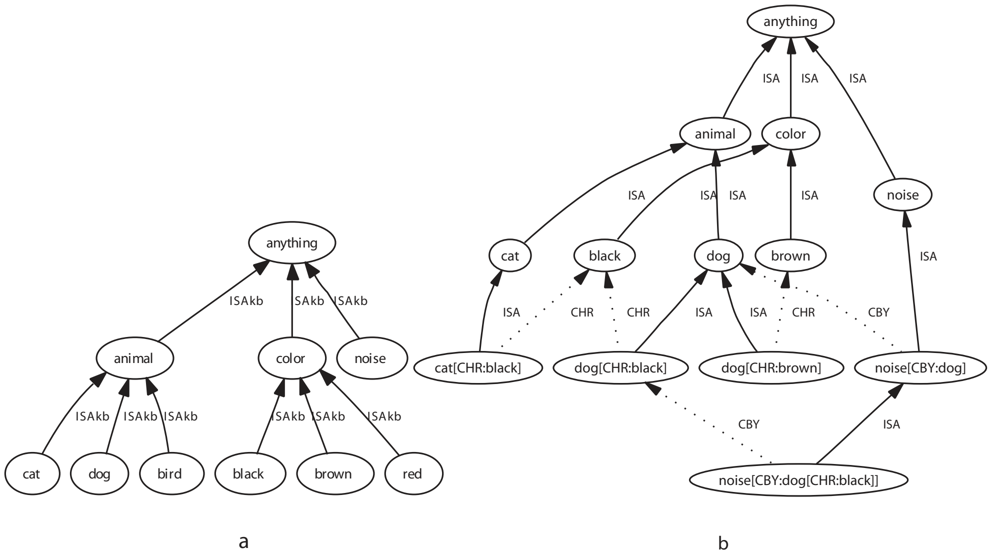


Figure 4.5: ***a****) An example of knowledge base ontology* ISA_{KB} ***b****) A simple instantiated ontology based on figure a and the set of instantiated concepts* $cat[\text{CHR}\!:\!black]$, $dog[\text{CHR}\!:\!black]$, $dog[\text{CHR}\!:\!brown]$, $noise[\text{CBY}\!:\!dog[\text{CHR}\!:\!black]]$.

For a set of terms we define $\tau(C) = \bigcup_{c\epsilon C} \tau(c)$. For example:

$$\tau(c_1[r_1\colon c_2[r_2\colon c_3]]) \quad = \quad \{c_1[r_1\colon c_2[r_2\colon c_3]], c_1[r_1\colon c_2], c_1, c_2[r_2\colon c_3], c_2, c_3\}.$$

Let $\omega(C)$ for a set of terms C be the transitive closure of C with respect to $\leq$. Then the expansion of the set of instantiated concepts I becomes:

$$\widehat{I} = \omega(\tau(I))$$

Now, the C-restiction sub-ontology $O_C = (C, \leq, \mathbf{R})$ with respect to a given set of concepts C, is the sub-ontology of O over concepts in C connected by $\leq$ and $\mathbf{R}$. Thus the instantiated ontology $O_{\widehat{I}} = (\widehat{I}, \leq, \mathbf{R}) = (\omega(\tau(I)), \leq, \mathbf{R})$ is the $\widehat{I}$-restiction sub-ontology of O.

Finally, ISA is defined as the transitive reduction of $\leq$ and consider ($\widehat{I}$,ISA, $\mathbf{R}$) for a visualization and as the basis for similarity computation.

Consider the knowledge base ontology ISA_{KB} shown in Figure 4.5a. In this case:

$$\mathbf{A} \quad = \quad \{cat, dog, bird, black, brown, red, animal, color, noise, anything\}$$

and $\mathbf{L}$ includes $\mathbf{A}$ and any combination of compound terms combining elements of $\mathbf{A}$ with attributes from $\mathbf{A}$ by relations from $\mathbf{R}$, due to the generative quality of the ontology.

Now assume a miniature target document collection with the following instantiated concepts:

$$\begin{aligned} I &= cat[\text{CHR}{:}black], dog[\text{CHR}{:}black], dog[\text{CHR}{:}brown], \\ &\quad noise[\text{CBY}{:}dog[\text{CHR}{:}black]] \end{aligned}$$

The decomposition $\tau(I)$ includes any sub-term of elements from I, while $\widehat{I} = \omega(\tau(I))$ adds the subsuming $\{animal, color, anything\}$:

$$\begin{aligned} \widehat{I} &= \{cat, dog, black, brown, animal, color, noise, anything, \\ &\quad cat[\text{CHR}{:}black], dog[\text{CHR}{:}black], dog[\text{CHR}{:}brown], \\ &\quad noise[\text{CBY}{:}dog], noise[\text{CBY}{:}dog[\text{CHR}{:}black]]\} \end{aligned}$$

where the concepts *red* and *bird* from $\mathbf{A}$ are omitted because they are not instantiated.

The resulting instantiated ontology $(\widehat{I}, \leq, \mathbf{R})$ is transitively reduced into the domain-specific ontology $(\widehat{I}, \text{ISA}, \mathbf{R})$, as shown in Figure 4.5b.

An instantiated ontology can describe the domain of any given subset of concepts with respect to some resource ontologies, spanning from a complete document base to a single concept.

4.4 Summary and Discussion

The issues for discussion in this chapter are all related to what is denoted as *ontological indexing.* This chapter began by using the idea of representation established in Chapter 3 as its basis.

In order to discuss different kinds of description notations/languages two important properties of descriptions, exhaustivity and specificity, were described, where exhaustivity is a property of index descriptions and specificity is a property of descriptors. Furthermore, two different views on the categorization of descriptions have been defined, fidelity and the notion of high and low level descriptions, where fidelity refers to the closeness of the descriptions to the content of what they describe, and high and low level refer to the internal content of descriptions. These properties and measures have been used throughout the chapter to describe the different kinds of descriptions that have been used in order to reach the preferred description, a collection of Ontolog expressions.

In the discussion of the two properties, exhaustivity and specificity, we saw that they could be defined either conceptually or statistically. The conceptual definition harmonizes perfectly with the idea of ontology-based retrieval. The statistical definition is naturally precisely as relevant for ontology-based retrieval as it is for lexical-based retrieval. In the conceptual definition of specificity, "*private school*" is considered more specific than "*school*" since the former is subsumed by the latter. The statistical notion of specificity can be seen in relation to the inverse document frequency in term weighting (see Section 2.1.1), where a term attached to most documents is considered less specific than one attached to few documents. This statistical notion would therefore still be interesting combined with the conceptual notions as they would refer to specificity in two dimensions, since a conceptually specific term, e.g. "*private school*", could be attached to many documents in a narrow domain.

In order to establish ontology based indexing, we have to reveal the semantics of a text, i.e. documents and queries. As a result, natural language processing has to be enclosed in the indexing.

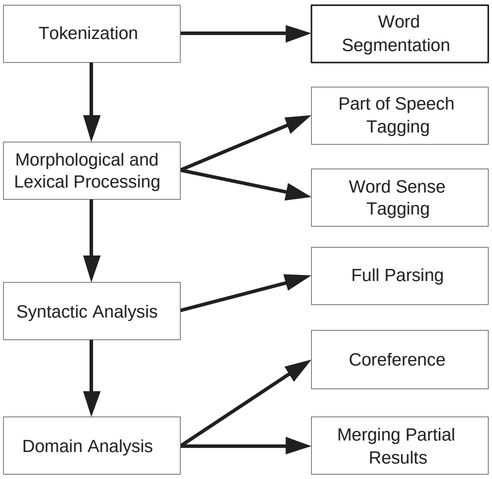


Figure 4.6: *The modules of partial semantic parsing in information extraction, from Appelt and Israel [1999].*

The modules in the natural language processing described in this chapter are the ones commonly used for partial parsing. Figure 4.6 illustrates partial parsing as conceived in information extraction[3]; a description of the left-hand side modules is given in terms of what they involve on the right-hand side.

The semantic analysis described in this chapter looks at approaches for the disambiguation of word classes and senses. An alternative model, where the ambiguities are weighted proportionally to the probability of occurrence, could be an alternative solution. This would naturally require methodologies that support ambiguities, where the selection of a particular word class and sense can be postponed until descriptions are generated or can even be avoided, thus leaving ambiguities open for the query evaluation. Obviously, one of the advantages of disambiguation is a reduction in the computational complexity, since parse trees are reduced through the different modules in the natural

[3] Information extraction is a field between computational linguistics and information retrieval, defined as the process of taking unseen texts as input and producing fixed-format, unambiguous data as output.

language processing. On the other hand, any wrong choice in the process would set off a cascade of errors into the succeeding modules and probably prevent them from making the right choice.

An interesting discussion about the recognition of the relational connection between concepts is the upper boundary for compound concepts. Using indexing on the sentence level, imagine the possibility of parsing and generating only one compound concept per sentence, namely the expression which describes the full meaning of the sentence. From a linguistic point of view this would probably be a preferable goal, but for the information retrieval process it would introduce some problems. In order to support quick retrieval, one important goal would be the ability to compare descriptions without reasoning. The only scalable solution to this is a comparison of items which can be optimized with database indexes for very quick retrieval. It is therefore necessary to somehow transform the information of the compound concepts into a set of strings. One such method is by expansion of the compound concept through the ontology into a set of similar concepts. Using simple string comparison, it would then be possible to search for all documents with one or more of the "strings" in the attached expansion. The similarity between concepts could then be measured by the distance in the ontology, which is looked at in the following chapter. Another method would mean splitting the compound concepts by using the decomposition function τ. Take, for instance, the concept of $dog[\textsc{chr}{:}large]$:

$$\tau(dog[\textsc{chr}{:}large]) = \{dog[\textsc{chr}{:}large], dog, large\}$$

Note that the consequence of this is similar to adding the compound concept into the nested set representation used in the first version of the OntoQuery prototype, and so introduces descriptions which are nested sets of Ontolog expressions. The cardinality of the intersection between decompositions could then express the similarity between concepts, and the comparison would then be string-based. Since the compound concept itself is a member of the decomposition of only that particular concept or a more specialized concept subsumed by it, it could possibly have the same members.

The maximum level of descriptions in a given retrieval system has to be determined empirically, since it has to be balanced between the "cost" with respect to the extra time used during the query evaluation and the refinement of the result of the querying.

This scalable level of descriptions with respect to the "compoundness" of the descriptions can be seen as a scaling of the semantic analysis. This could be useful, for instance, as a parameter in the querying process in order to broaden or narrow the result, since the level of compoundness is related to the specificity of the descriptions. Another possible use of this kind of scalability is in connection with smaller document bases or the retrieval of information inside a single document, where time may have a minor influence due to the smaller size, or the fact that the task is more well-understood, which means a higher level of specificity may be preferred.

Finally, instantiated ontologies are presented. They serve as a way to restrict a given general ontology to a set of instantiated concepts. An instantiated ontology of this type could, for example, be an ontology of the concepts in a given query that could in turn serve as a solution to disambiguation if the user were able to choose between different senses in an instantiated ontology of their queries. Another possible use is as a visualization of the similarity between documents by showing how the instantiated ontologies of the concepts they are described by overlap (see Section 6.5).

Chapter 5

Ontological Similarity

In this chapter we focus on the introduction of ontology-based similarity in information retrieval. The obvious use of ontological similarity in information retrieval is as a replacement of the conventional lexical equivalence. Instead of retrieving documents solely on the basis of the occurrence of query terms, the documents containing terms that are semantically related to the query terms could be taken into consideration [Cohen and Kjeldsen, 1987; Rada and Bicknell, 1989].

The indexing process maps information found in documents into the ontology, identifying concepts and their positions in the ontology. Information in queries can similarly be mapped into the ontology, and thus in addition to retrieving the exact match (the documents which have ALL concepts from the query assigned), the structure of the ontology can be used to retrieve semantically related documents. Naturally, whenever queries contain more than one piece of information (one concept) some kind of aggregation is needed to compute the retrieval, which is the topic of the next chapter. In this chapter, the focus is on *ontological concept similarity*, the similarity between concepts in ontologies.

The problem of formalizing and quantifying the intuitive notion of *semantic similarity* between lexical units has a long history in philosophy, psychology, and artificial intelligence, going back at least to Aristotle [Budanitsky, 1999]. Among the heralds of the contemporary wave of research are Osgood [1952], Quillian [1968], and Collins and Loftus [1988]. Osgood's "semantic differential" was an attempt to measure similarity as the Euclidian distance in an n-dimensional space. Qullian, Collins and Lofus focused on "spreading activation", which measures similarity using distances in semantic networks. The former was rejected by Osgood himself as he found his system relied on "connotative emotions" rather than "denotative meaning", while the idea of spreading activation, on the other hand, still motivates researchers in lexical semantics.

Similarity, *relatedness* and *distance* are three different terms used in the literature, sometimes interchangeably, when referring to the topic of this chapter. Some attempts have been made to differentiate between these terms, for instance, by Resnik [1995], who defines *similarity* as a special case of *relatedness*. This thesis does not differentiate between *similarity* and *relatedness* and uses the two terms interchangeably, or in such a way that the context communicates the exact meaning. With *distance*, on the other hand, there is clearly a duality, where high similarity implies low distance and vice versa.

The foundation for measuring similarity in this thesis is ontologies formed by a set of concepts interrelated by a set of semantic relations. The concept inclusion relation, ISA, is defined as the ordering relation. The ISA relation is normally defined as a transitive relation in ontologies, hence, if A ISA B and B ISA C, then A ISA C, e.g. from WordNet[1] "plankton" ISA "organism" and "organism" ISA "living thing", then also "plankton" ISA "living thing". In order to use the structure of ontologies for measuring similarity as the distance between concepts, a non-transitive ISA relation is required, otherwise the shortest *distance* between, for instance, "plankton" and "living thing", and "organism" and "living thing" from the above example, for that matter, is the same as it would be for any super-ordinate concept. All the relations referred to in this chapter are therefore considered as non-transitive relations in transitively reduced ontologies, unless explicitly defined otherwise.

In this chapter we present and discuss different similarity measures based on ontologies. Initially, a set of basic intuitive properties are defined to which the adherence of similarity measures in information is preferable. Some additional properties are revealed as we introduce the measures and finally, there is a set of properties categorized as *basic properties*, *retrieval-specific properties*, and *structure-specific properties*. For each of the measures presented, a discussion is presented of how they comply, partially or fully, with these intuitive expectations. In the last part of this chapter, an experiment is performed in which similarity measures are compared with human similarity judgments.

5.1 Path Length Approaches

In this section, two simple approaches are presented that are based on taxonomies that measure similarity through the concept inclusion relation.

Before presenting these measures, we introduce some very *basic properties* originally defined by Lin [1997; 1998] in his attempt to create a measure of similarity that was both universally applicable to arbitrary objects and theoretically justified, and hence not tied to particular applications, domains, resources or a specific knowledge representation.

[1] WordNet version 2.1.

Property 1: Commonality Property
The similarity between A and B is related to their commonality. The more commonality they share, the more similar they are.

Property 2: Difference Property
The similarity between A and B is related to the differences between them. The more differences they have, the less similar they are.

Property 3: Identity Property
The maximum similarity between A and B is reached when A and B are identical, no matter how much commonality they share.

Any similarity measure must necessarily comply with these properties as they express the abstract notion of similarity.

Furthermore, we initially define an important *retrieval-specific property.* Common similarity measures are also symmetric: $sim(A, B) = sim(B, A)$. If we just consider similarity between concepts, then the order of these is normally not taken into account, e.g. the similarity between "*a cat*" and "*a dog*" would be considered equal to the similarity between "*a dog*" and "*a cat*". However, in a context where the order between concepts can be determined by a taxonomy or an ontology, the symmetry property becomes deceptive, e.g. the ontology in Figure 5.1 implies that *sim(D,B)* $<$ *sim(B,D).*

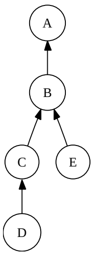


Figure 5.1: *The property of generalization implies that sim(D,B)* $<$ *sim(B,D)*

That symmetry should not apply for a measure of similarity is also supported by Tversky [1977], who argues that similarity judgments can be regarded as extensions of similarity statements, i.e. statements of the form "a is like b", which are obviously directional. He gives a number of examples: "the portrait resembles the person", and not "the person resembles the portrait", "the son resembles the father", and not "the father resembles the son".

Consider an ontology where "*plankton*" is a specialization of "*organism*", then the intuition is that "*plankton*" satisfies the intension of a query on "*organism*", whereas "*organism*" (which could be of any kind) does not necessarily satisfy the intention of a query on "*plankton*" [Andreasen *et al.*, 2003b; Andreasen *et al.*, 2003a].

This indicates that in the ontology-based information retrieval context, the similarity measure cannot be symmetrical, and should somehow capture that the "cost" of similarity in the direction of the inclusion (generalization) should be significantly higher than the similarity in the opposite direction of the inclusion (specialization).

Property 4: Generalization Property
Concept inclusion implies reduced similarity in the direction of the inclusion.

5.1.1 Shortest Path Length

One obvious way to measure similarity in a taxonomy, given its graphical representation, is to evaluate the distance between the nodes corresponding to the items being compared, where a shorter distance implies higher similarity.

In Rada et al. [1989] a simple approach based on the shortest path length is presented. The principal assumption is that the number of edges between terms in a taxonomy is a measure of conceptual distance between concepts:

$$dist_{Rada}(c_i, c_j) = \text{minimal number of edges in a path from } c_i \text{ to } c_j \qquad (5.1)$$

Surprisingly good results can be obtained using this approach, despite its simplicity. One of the reasons for this is that when the paths are restricted to ISA relations, the shortest path length corresponds to the conceptual distance [Budanitsky, 2001]. Another reason for good reported results is probably the specificity of the domain used in Rada's experiments (the Medical Subject Headings (MeSH)), which ensures the relative homogeneity of the hierarchy.

Among the set of intuitive properties defined so far, only the basic properties, commonality, difference and identity are true for this measure. Observe that the measure of the shortest path length is a distance function and thereby a metric. Thus the measure is obviously in accordance with the commonality and difference properties, and the identity property corresponds to the zero property of a metric.

The shortest path length measure does not comply with the generalization property because the measure is symmetric.

5.1.2 Weighted Shortest Path

In Bulskov et al. [2002], another simple edge-counting approach is presented[2]. It is argued that concept inclusion (ISA) intuitively implies strong similarity in the opposite direction of inclusion (specialization). In addition, the direction of the inclusion (generalization) must contribute some degree of similarity, as,

[2] The presentation of Weighted Shortest Path measure in Section 5.1.2 is a rendering of the original presentation in [Bulskov *et al.*, 2002].

for example, in the small excerpt of an ontology in Figure 5.2. With reference to this ontology, the atomic concept *dog* has high similarity to the concepts *poodle* and *alsatian*.

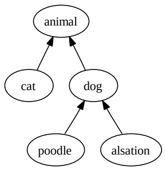


Figure 5.2: *An example ontology covering pets*

The measure respects the ontology in the sense that every concept subsumed by the concept *dog* by definition bears the relation ISA to *dog*. The intuition is that the answer to a query on *dog* including the instance *poodle* is satisfactory (a specific answer to a general query). Because the ISA relation obviously is transitive, the same argument can be used to include further specializations, e.g. including *poodle* in the extension of *animal*. However, similarity exploiting the taxonomy should also reflect the *distance* in the relation. Intuitively, greater distance (longer path in the relation graph) corresponds to smaller similarity.

Furthermore, generalization should contribute to similarity. Though, of course, this is not strictly correct, because all *dogs* are *animals* and *animals* are to some degree similar to *dogs*. Thus, the property of generalization similarity should be exploited. However, for the same reasons as in the case of specializations, transitive generalizations should contribute with a decreased degree of similarity.

A concept inclusion relation can be mapped into a similarity function in accordance with the two properties described above as follows. Assume an ontology given as a domain knowledge relation. Figure 5.3 can be viewed as such an example. To make *distance* influence similarity, we assume the ISA relation to be transitively reduced.

Similarity reflecting "distance" can then be measured from the path length in the graph corresponding to the ISA relation. By parameterizing with two factors, $\sigma \in [0,1]$ and $\gamma \in [0,1]$, which express similarity of immediate specialization and generalization respectively, a simple similarity function can be defined as follows. A path between nodes (concepts) x and y using the ISA relation:

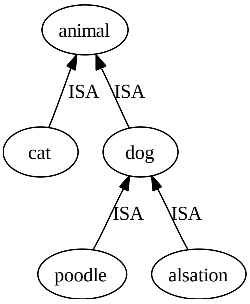


Figure 5.3: *An example ontology with relation* ISA *covering pets*

$$P = (p_1, \cdots, p_n)$$

where

$$p_i \text{ ISA } p_{i+1} \text{ or } p_{i+1}\text{ISA } p_i$$

for each i with $x = p_1$ and $y = p_n$.

Given a path $P = (p_1, \cdots, p_n)$, set $s(P)$ to the number of specializations and $g(P)$ to the number of generalizations along the path P, as follows:

$$s(P) = |\{i|p_i \text{ ISA } p_{i+1}\}| \tag{5.2}$$

and

$$g(P) = |\{i|p_{i+1} \text{ ISA } p_i\}| \tag{5.3}$$

If $P^1, \cdots, P^m$ are all paths connecting x and y, then the degree to which y is similar to x can be defined as follows:

$$sim_{WSP}(x, y) = \max_{j=1,\ldots,m} \left\{ \sigma^{s(P^j)} \gamma^{g(P^j)} \right\} \tag{5.4}$$

We denote that this measure, $sim(x, y)_{WSP}$ (Weighted Shortest Path), as the similarity between two concepts x and y, is calculated as the maximal product of weights along the paths between x and y.

This similarity can be considered as derived from the ontology by transforming the ontology into a directional weighted graph with σ as downward weights and γ as upward weights, and with similarity derived as the product of the weights on the paths. Figure 5.4 shows the graph corresponding to the ontology in Figure 5.3.

The *weighted shortest path* measure is a generalization of the *shortest path length* measure and would therefore hold for the two basic properties commonality and difference, too. Since $\sigma \in [0, 1]$ and $\gamma \in [0, 1]$ maximum of $\sigma^{s(P^j)}$ and

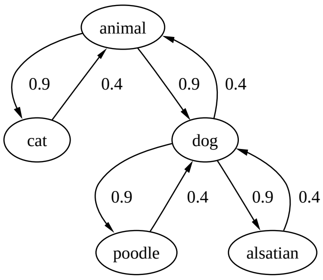


Figure 5.4: *The ontology transformed into a directed weighted graph with the immediate specialization and generalization similarity values $\sigma = 0.9$ and $\gamma = 0.4$ as weights. Similarity is derived as the maximum (multiplicative) weighted path length, and thus $sim(poodle, alsatian) = 0.4 * 0.9 = 0.36$.*

$\gamma^{g(P^j)}$ is obtained when $s(P^j) = 0$ and $g(P^j) = 0$. Thus the identity property is obeyed by the *weighted shortest path* measure. Furthermore, the measure is in accordance with the generalization property, due to the weighted edges.

5.2 Depth-Relative Approaches

Despite the apparent simplicity, the edge counting approaches have a widely acknowledged problem of typically relying on edges in the taxonomy to represent uniform distances. Consider the two pairs of concepts taken from WordNet 1) "pot plant" and "garden plant" and 2) "physical entity" and "abstract entity". Using our intuition, we would judge the similarity for the first pair to be higher than the similarity for the second, since the first pair of concepts is much more specific [Sussna, 1993]. This means that the distance represented by an edge should be reduced with an increasing depth (number of edges from the top) of the location of the edge, which leads to the first structure-specific property:

> **Property 5: Depth Property**
> The distance represented by an edge is influenced by the depth of the location of the edge in the ontology.

In the *weighted shortest path*, edges contribute non-uniform distances; a step along a generalization edge is longer than a step along a specialization edge. However, this non-uniformity does not resemble the depth property, since these weights are assigned independently of the depth of the taxonomy or ontology, because the weights are defined for generalizations and specializations in general.

The approaches presented in this section, *depth-relative scaling*, *conceptual similarity*, and *normalized path length*, are basically *shortest path length* approaches which take into account the depth of the edges connecting two concepts in the overall structure of the ontology. All these approaches comply therefore with the same properties as the *shortest path length* and the newly defined depth property. Obviously, these measures do not comply with the generalization property as they are symmetric.

5.2.1 Depth-Relative Scaling

In his depth-relative scaling approach [1993], Sussna defines two edges representing inverse relations for each edge in a taxonomy. The weight attached to each relation r is a value in the range $[\min_r; \max_r]$. The point in the range for a relation r from concept c_1 to c_2 depends on the number n_r of edges of the same type, leaving c_1, which is denoted as the *type specific fanout* factor:

$$w(c_1 \rightarrow_r c_2) = \max_r - \frac{\max_r - \min_r}{n_r(c_1)}$$

which, according to Sussna, reflects the dilution of the strength of the connotation between the source and the target concept. The two inverse weights are averaged and scaled by depth d of the edge in the overall taxonomy, which is motivated by the observation that sibling-concepts deeper in the taxonomy appear to be more closely related than those higher in the taxonomy. The distance between adjacent nodes c_1 and c_2 are computed as:

$$\mathrm{dist}_{sussna}(c_1, c_2) = \frac{w(c_1 \rightarrow_r c_2) + w(c_2 \rightarrow_{r\prime} c_1)}{2d} \tag{5.5}$$

where r is the relation that holds between c_1 and c_2, and $r\prime$ is its inverse.

The semantic distance between two arbitrary concepts c_1 and c_2 is computed as the sum of distances between the pairs of adjacent concepts along the shortest path connecting c_1 and c_2.

5.2.2 Conceptual Similarity

Wu and Palmer propose a measure of semantic similarity in their paper [1994] on the semantic representation of verbs in computer systems and its impact on lexical selection problems in machine translation. Wu and Palmer define *conceptual similarity* between a pair of concepts c_1 and c_2 as:

$$sim_{Wu\&Palmer}(c_1, c_2) = \frac{2 \times N3}{N1 + N2 + 2 \times N3} \tag{5.6}$$

where $N1$ is the number of nodes on a path from c_1 to a concept c_3, denoting the least upper bound of both c_1 and c_2. $N2$ is the number of nodes on a path from c_2 to c_3. $N3$ is the number of nodes from c_3 to the most general concept (the topmost node in the tree).

5.2.3 Normalized Path Length

Leacock and Chodorow [1998] proposed an approach for measuring semantic similarity as the shortest path using ISA hierarchies for nouns in WordNet. The different noun hierarchies are combined into a single hierarchy by introducing a topmost node, subsuming all the topmost nodes in all the noun hierarchies[3]. This ensures the existence of a path between all synsets in the taxonomy.

The proposed measure determines the semantic similarity between two synsets (concepts) by finding the shortest path and by scaling using the depth of the taxonomy:

$$sim_{Leacock\&Chodorow}(c_1, c_2) = -\log\left(\frac{Np(c_1, c_2)}{2D}\right)$$

where c_1 and c_2 represents the two concepts, $Np(c_1, c_2)$ denotes the shortest path between the synsets (measured in nodes), and D is the maximum depth of the taxonomy.

5.3 Corpus-Based Approaches

The similarity measures presented so far use knowledge solely captured by the ontology (or taxonomy) to compute a measure of similarity. In this section, we present three approaches that incorporate corpus analysis as an additional, and qualitatively different knowledge source. The knowledge revealed by the corpus analysis is used to augment the information already present in the ontologies or taxonomies.

5.3.1 Information Content

Resnik [1999] argued that a widely acknowledged problem with edge-counting approaches was that they typically rely on the notion that edges represent uniform distances. One criterion of similarity between two concepts is the extent to which they share information, which for a taxonomy can be determined by the relative position of their least upper bound. This criterion seems to be captured by edge-counting approaches, for instance, the *shortest path length* approach [Rada *et al.*, 1989] presented above. However, the edge-counting approaches in general do not comply with the depth property, since edges typically represent uniform distances and the position in the hierarchy of the least upper bound is not taken into account. In the last section, several examples of edge-counting measures are presented that compensate for this problem by using the depth in the hierarchy in measuring the similarity of the concepts

[3] Remember that WordNet does not have one unique topmost node, but 25 unique beginners instead. (see Section 3.5.3)

being compared. Resnik's measure, *information content*, uses knowledge from a corpus about the use of senses to express non-uniform distances.

Let C denote the set of concepts in a taxonomy that permits multiple inheritance and associates with each concept $c \in C$, the probability $p(c)$ of encountering an instance of concept c. Following the standard definition from Shannon and Weaver's information theory [1949], the *information content* of c is then $-\log p(c)$. For a pair of concepts c_1 and c_2, their similarity can be defined as:

$$sim_{Resnik}(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log(p(c))] \tag{5.7}$$

where $S(c_1, c_2)$ is the set of least upper bounds in the taxonomy of c_1 and c_2. $p(c)$ is monotonically non-decreasing as one moves up in the taxonomy, and if c_1 ISA c_2 then $p(c_1) \leq p(c_2)$.

Given the formula in (5.7), the similarity between two words w_1 and w_2 can be computed as:

$$wsim_{resnik}(w_1, w_2) = \max_{c_1 \in s(w_1), c_2 \in s(w_2)} [sim(c_1, c_2)] \tag{5.8}$$

where $s(w_i)$ defines the set of possible senses for the word w_i.

Resnik describes an implementation based on *information content* using WordNet's [Miller, 1990] taxonomy of noun concepts [1999]. The information content of each concept is calculated using noun frequencies from the Brown Corpus of Standard American English [Francis and Kucera, 1964].

Each noun occurring in the corpus was counted as an occurrence of each taxonomic class that contained it. For example, in Figure 5.5, an occurrence of the noun *dime* would increment the frequency of *dime*, *coin*, *cash*, and so forth:

$$freq(c) = \sum_{n \in words(c)} count(n),$$

where $words(c)$ is the set of words whose senses are subsumed by concept c, and adopt the maximum likelihood estimate:

$$\hat{p}(c) = \frac{freq(c)}{N}$$

where N is the total number of nouns.

The *information content* approach depends completely on the least upper bound and is therefore basically a *shortest path* approach and would therefore also comply with the set of basic properties, and disobey the generalization property as it is symmetric. Due to additional knowledge from the corpus analysis, the *information content* approach obeys the depth property.

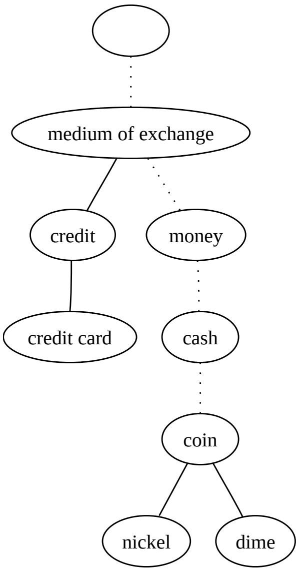


Figure 5.5: *Fragment of the WordNet taxonomy. Solid edges represent* ISA*; dotted links indicate that concepts were removed to save space.*

5.3.2 Jiang and Conrath's Approach

The idea of the approach suggested by Jiang and Conrath [1997] was to synthesize edge-counting methods and information content into a combined model by adding the latter as a corrective factor.

The general formula for the edge weight between a child concept c_c and a parent concept c_p by considering factors such as local density in the taxonomy, node depth, and link type is:

$$wt(c_c, c_p) = \left(\beta + (1-\beta)\frac{\bar{E}}{E(c_p)}\right)\left(\frac{d(c_p)+1}{d(c_p)}\right)^{\alpha} LS(c_c, c_p)T(c_c, c_p),$$

where $d(c_p)$ is the depth of the concept c_p in the taxonomy, $E(c_p)$ is the number of children of c_p (the local density), $(\bar{E})$ is the average density in the entire taxonomy, $LS(c_c, c_p)$ is the strength of the edge between c_c and c_p, and $T(c_c, c_p)$ is the edge relation/type factor. The parameters $\alpha, \alpha \geq 0$ and $\beta, 0 \leq \beta \leq 1$ control the influence of concept depth and density, respectively.

In the framework of a taxonomy, Jiang and Conrath argued that the strength of a link $LS(c_c, c_p)$ between parent and child concepts is proportional to the conditional probability $p(c_c|c_p)$ of encountering an instance of the child concept, c_c, given an instance of the parent concept, c_p:

$$LS(c_c, c_p) = -\log p(c_c|c_p)$$

by definition:

$$p(c_c|c_p) = \frac{p(c_c \cap c_p)}{p(c_p)}$$

It follows from Resnik's way of assigning probabilities to concepts that $p(c_c \cap c_p) = p(c_c)$, because any instance of a child concept c_c is also an instance of the parent concept c_p. Then:

$$p(c_c|c_p) = \frac{p(c_c)}{p(c_p)}$$

and

$$LS(c_c, c_p) = IC(c_c) - IC(c_p),$$

if $IC(c)$ denotes the information content of concept c.

Jiang and Conrath then defined the semantic distance between two nodes as the summation of edge weights along the shortest path between them [J. Jiang, 1997]:

$$dist_{Jiang\&Conrath}(c_1, c_2) = \sum_{c \in \{path(c_1,c_2) - LSuper(c_1,c_2)\}} wt(c, parent(c)),$$

where $path(c_1, c_2)$ is the set of all nodes along the shortest path between concepts c_1 and c_2, $parent(c)$ is the parent node of c and $LSuper(c_1, c_2)$ is the lowest superordinate (least upper bound) on the path between c_1 and c_2. The reason for its removal from the set is that it has no parent within the set.

Jiang and Conrath's approach complies as *information content* to the basic properties and the depth property, but not to the generalization property.

5.3.3 Lin's Universal Similarity Measure

Lin [1997; 1998] defines a measure of similarity claimed to be both universally applicable to arbitrary objects and theoretically justified. Upon recognizing that known measures generally are tied to a particular application, domain, or resource, he encourages the need for a measure that does not presume a specific kind of knowledge representation and that is derived from a set of assumptions, rather than directly from a formula.

The formal definition for Lin's information-theoretic definition of similarity builds on three basic properties, commonality, difference and identity. In addition to these, Lin introduces a few other assumptions and definitions, notably that the commonality between A and B is measured by the amount of information contained in the proposition that states the commonalities between them, formally:

$$I(common(A, B)) = -\log p(common(A, B)),$$

where the information $I(s)$ contained in a proposition s is measured as the negative logarithm of the probability of the proposition, as described by Shannon [1949].

The difference between A and B is measured by:

$$I(description(A, B)) - I(common(A, B)),$$

where $description(A, B)$ is a proposition about what A and B are.

Given the above setting and the apparatus described in *Information Theory* [Shannon and Weaver, 1949], Lin was able to prove that the similarity between A and B is measured by the ratio between the amount of information needed to state the commonality of A and B and the information needed to describe fully what they are:

$$sim_{Lin}(A, B) = \frac{\log p(common(A, B))}{\log p(description(A, B))}.$$

His measure of similarity between two concepts in a taxonomy ensures that:

$$sim_{Lin}(c_1, c_2) = \frac{2 \times \log p(\text{LUB}(c_1, c_2))}{\log \quad p(c_1) + \log p(c_2)},$$

where $\text{LUB}(c_1, c_2)$ is the least upper bound of c_1 and c_2, and where $p(x)$ can be estimated based on statistics from a sense tagged corpus (for example Resnik's *information content*, see Section 5.3.1).

The *universal similarity measure* does not comply differently with the set of properties than the *information content* and the *combined approach*, and would therefore comply with the set of basic properties and the depth property, but would fail to comply with the generalization property as it is symmetric.

5.4 Multiple-Paths Approaches

In the measures presented up to this point, only one path between concepts, the shortest, contributes to their similarity. Since all these measures are based on taxonomies, the relation in consideration is the concept inclusion relation. In this section, measures are presented that take into account all semantic relations in ontologies. An approach taken by some of these measures is to take into consideration more than one path between concepts. To distinguish these, a final structure property is introduced to express this approach.

Intuitively, attributes should influence the measure of similarity, thus allowing two concepts sharing the same attribute to be considered as more similar, compared to concepts not having this particular attribute. For instance, the definition "*blue vehicles*" would cluster vehicles by an attribute and allow all vehicles not sharing this attribute to be considered less similar when compared to a vehicle from this cluster, i.e. $sim(car[\textsc{chr}{:}blue], truck[\textsc{chr}{:}blue]) > sim(car[\textsc{chr}{:}blue], truck[\textsc{chr}{:}big])$.

Another problem is that a similarity measure that finally selects, independently of whether or not all semantic relations are considered, one path as a measure for the similarity, fails to truly express similarity whenever the ontologies allow multiple inheritance. Naturally, in some cases, only one of the inherited senses actually influences the similarity measure. Limiting a similarity measure to only one path is in contrast to the idea of multiple inheritance, since such concepts cannot be described solely by their inheritance from only one of their superordinate concepts.

> **Property 6: Multiple-Paths Property**
> The similarity between concepts is related to the number of paths connecting the concepts and the length of these paths.

The multiple-paths property concerns the inclusion of more than "the best path" in the measure. Apart from accounting for multiple paths combining taxonomic and semantic relations, compliance to this property may also influence purely taxonomic-based similarity based on multiple inheritance ontologies.

5.4.1 Medium-Strong Relations

Hirst and St-Onge [Hirst and St-Onge, 1998; St-Onge, 1995] distinguished three major kinds of relations between nouns in WordNet: extra-strong, strong and medium-strong relations. The extra-strong relation is only between a word and its literal repetition. A strong relation between two words exists if:

1. They have a synset in common;
2. There is a horizontal link (antonymy, similarity) between a synset of each word; or
3. There is any kind of link at all between a synset of each word or if one word is a compound or phrase that includes the other word.

Medium-strong, the final type of relation, exists between words when a member of a set of allowable paths connects the two words. A path is allowable if it contains no more than five edges and conforms to one of the eight patterns described in Hirst and St-Onge [1998]. The weight of a path is expressed by the following formulae:

$sim_{Hirst\&St\text{-}Onge}(c_1, c_2) = C - path\ length - k * number\ of\ changes\ in\ direction,$

where C and k are constants. Thus, the longer the path and the more changes in direction, the lower the weight.

The *medium-strong relation* is basically a shortest path length measure and thus does not comply with the multiple-path property; hence even through it introduces both taxonomic and semantic relations, it is still restricted to only one path. It does not comply with either the generalization or depth properties, but obviously obeys the basic properties as it is a *shortest path length* measure.

5.4.2 Generalized Weighted Shortest Path

The *weighted shortest path* presented earlier (see 5.1.2) can easily be refined to include the shortest path between concepts using all semantic relations[4]. While the similarity between $c[r_1 \colon c_1]$ and c can be claimed to be justified by the ontology formalism (subsumption), it is not strictly correct in an ontological sense to claim $c[r_1 \colon c_1]$'s similarity to c_1.

For instance, *noise*[CBY: : *dog*] is conceptually not a type of *dog*. On the other hand, it would be reasonable to claim that *noise*[CBY: : *dog*], in a broad sense, has something to do with a dog, and thus has similarities to *dog*. Considering a wider number of semantic relations allows the option of calculating a more finely-grained similarity between concepts.

Consider Figure 5.6. The solid edges are ISA references and the broken ones are references by other semantic relations. In the figure, CBY and CHR are used, denoting "*caused by*" and "*characterized by*", respectively. Each compound concept has broken edges to its attribution concept.

The principle of weighted path similarity can be generalized by introducing similarity factors for the semantic relations. However, there does not seem to be an obvious way to differentiate based on direction. Thus, we can generalize simply by introducing a single similarity factor and simplify to bidirectional edges.

Assume that we have k different semantic relations $r^1, \ldots, r^k$ and let $\rho_1, \cdots, \rho_k$ be the attached similarity factors. Given a path $P = (p_1, \cdots, p_n)$, set $r^j(P)$ to the number of r^j edges along the path P thus:

$$r^j(P) = \left|\left\{ i \mid \; p_i \quad r^j \quad p_{i+1} \text{ for } 1 \leq i \leq n \; \right\}\right|, \tag{5.9}$$

where n is the number of concepts in the path P.

If $P^1, \cdots, P^m$ are all paths connecting c_1 and c_2 and where $s(P^j)$ and $g(P^j)$ are defined as in equations (5.2) and (5.3), respectively, then the degree to which y is similar to x can be defined as follows:

$$sim_{GWSP}(x, y) = \max_{j=1,\ldots,m} \left\{ \sigma^{s(P^j)} \gamma^{g(P^j)} {\rho_1}^{r^1(P^j)} \cdots {\rho_k}^{r^k(P^j)} \right\} \tag{5.10}$$

[4] The presentation of Generalized Weighted Shortest Path measure in Section 5.4.2 is a rendering of the original presentation in [Bulskov *et al.*, 2002].

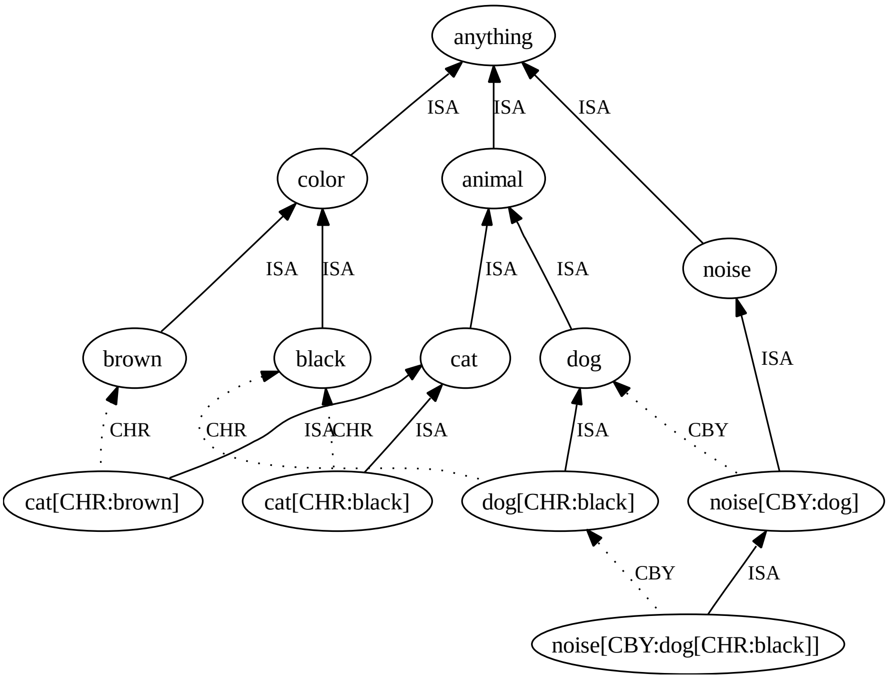


Figure 5.6: *An ontology where attribution with semantic relations is shown as dotted edges*

The result of transforming the ontology in Figure 5.6 is shown in Figure 5.7. Here, two semantic relations CHR and CBY are used. The corresponding edge-count functions are r^{WRT} and r^{CBY} and the attached similarity factors are denoted ρ_{WRT} and ρ_{CBY}. The figure shows the graph with the attached similarity factors as weights. Again, the degree to which concept c_1 is similar to concept c_2 is based on the shortest path.

For instance, we can derive from Figure 5.7 that $sim(cat, dog) = 0.9*0.4 = 0.36$ and $sim(cat[\text{CHR}::\ black], color) = 0.2 * 0.4 = 0.08$.

In contrast to normal weighting practice, which requires careful consideration by domain experts, the weights in the example were assigned in a rather ad hoc manner. The major difference between the *generalized weighted shortest path* approach and the *medium-strong relations* approach with respect to which properties they comply with is that the former obeys the generalization property.

5.4.3 Shared Nodes

All the approaches that have been presented until now only take into account one path in measuring similarity. Consequently, when two concepts are connected by multiple paths only one path, typically the shortest, contributes to

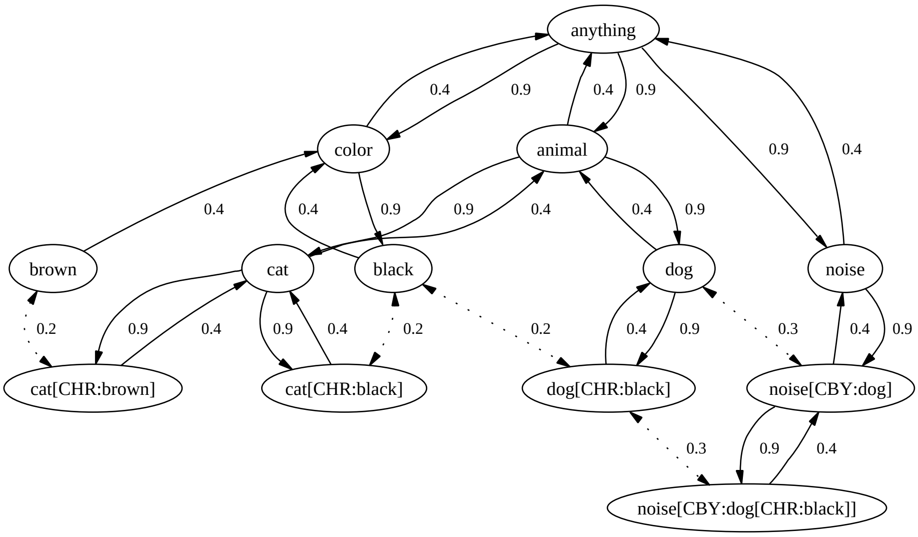


Figure 5.7: *The ontology of Figure 5.6 transformed into a directional weighted graph with the similarity factors: for specialization, $\sigma = 0.9$; for generalization, $\gamma = 0.4$; for* CBY:$\rho_{\text{CBY}} = 0.3$; *and for* CHR:$\rho_{\text{WRT}} = 0.2$

the similarity measure.

In order to make the sharing of attributes and multiple inheritance contribute to similarity, as argued for in the definition of the multiple-path property, then just a single path must be considered as the basis for measuring similarity.

One obvious approach for measuring similarity is to consider all possible connections between the concepts x and y[5]. Concepts, for instance, may be connected directly through inclusion and also through an attribute dimension, as $cat[CHR : black]$ and $poodle[CHR : black]$, or we might have multiple paths due to multiple inheritance. If the multiple connections phenomenon can be achieved without traversing all possible paths, we may have a more realistic means of similarity derivation.

The definitions of term decomposition used are:

$$\tau(c) = \{c\} \cup \{x | c \le x[\dots, r\colon y] \vee c \le y[\dots, r\colon x], x \in \mathbf{L}, y \in \mathbf{L}, r \in \mathbf{R}\}$$

and transitive closure of a set of concepts with respect to $\le$

$$\omega(C) = \{x | x \in C \vee y \in C, y \text{ ISA } x\}$$

from the definition of instantiated ontologies in Section 4.3. With $\alpha(x) = \omega(\tau(x))$ as the set of nodes (upwards) reachable from x in an instantiated

[5] The presentation of the Shared Nodes measure in Section is a slightly modified rendering of the original presentation in [Knappe *et al.*, 2005].

ontology, $\alpha(x) \bigcap \alpha(y)$ are the reachable nodes shared by x and y, which thus obviously is an indication of what x and y have in common. Immediate transformations of this into a normalized similarity measure are the fractions of the cardinality of the intersection and the cardinality of, respectively, the union $\alpha(x) \cup \alpha(y)$ and the individual $\alpha(x)$ and $\alpha(y)$, giving the following normalized measures:

(a)

$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x) \cup \alpha(y)|}$$

(b)

$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|}$$

(c)

$$sim(x, y) = \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|}$$

It is clear that instantiated ontologies and functions such as the above offer only a very coarsely-grained approximation of whatever the genuine similarity may be. In the discussion of *shared node* similarity functions below, the intuitive properties defined earlier are used to guide the choice of function.

First, it is important to note that the principle of using instantiated ontologies for deriving similarity unifies the concept inclusion relation with the semantic relations used in attribution. Not only *cat* but also *black* are understood to be related to *cat*[CHR:*black*], and *cat*[CHR:*black*] is understood to be related to *accident*[CBY:*cat*[CHR:*black*]].

The intuition of the generalization property is that, for instance, a *cat* satisfies the intention of an *animal*, whereas an *animal* (which could be of any kind) does not necessarily satisfy the intention of *cat*. From this property alone, the first alternative similarity function (a) above can be eliminated. A consequence of insisting on this property is that the similarity function cannot be symmetrical. In Figure 5.8, according to the generalization property, $sim(D, B) < sim(B, D)$.

Now consider alternative (c). Figure 5.8 shows that $sim(D, E) = \frac{2}{3}$ and $sim(E, D) = \frac{2}{4}$, which also violates the generalization property. Thus the only alternative that obeys this property is (b). In the example in Figure 5.8, $sim(D, E) = \frac{2}{4}$ and $sim(E, D) = \frac{2}{3}$ are obtained.

The depth property states, for instance, that siblings on low levels in the ontology, such as *pot plant* and *garden plant* should be higher than the similarity between siblings close to the top, such as *physical entity* and *abstract entity.* In our approach, when considering Figure 5.9, $sim(C, D) > sim(A, B)$, the

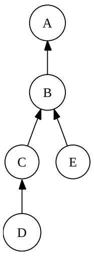


Figure 5.8: *The property of generalization implies that sim(D,B) < sim(B,D)*

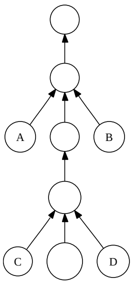


Figure 5.9: *The depth property implies that sim(C,D) > sim(A,B)*

similarity function (b) above satisfies the depth property, where $sim(A, B) = \frac{2}{3}$, while $sim(C, D) = \frac{4}{5}$ ((a) and (c) also satisfy this property).

Let the difference property be defined by distance in the ontology, then the similarity function (b) obviously does not satisfy this property because $sim(E, C) = sim(E, D)$. Hence, for K at any level of specialization below D, the result is still $sim(E, D) = sim(E, K)$.

To capture that further specialization implies reduced similarity, alternative similarity functions must be considered that are influenced by both specialization and generalization (like the function (a) above), but that still do not violate the generalization property. One modification that satisfies this is simply to take a weighted average of (b) and (c) above, as follows:

(d)

$$sim_{sharednodes}(x, y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1 - \rho) \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|},$$

where $\rho \in [0, 1]$ determines the degree of influence of the generalizations.

Although simplicity favors similarity (b), this measure, based on the aspects discussed, cannot be claimed to violate the semantics of the ontology, which means that similarity (d) still appears to be a better choice. Similarity (b) is simply a special case of sim_{sn} with $\rho = 1$.

The *shared nodes* approach with similarity function (d) complies with all the defined properties.

5.4.4 Weighted Shared Nodes Similarity

Intuition tells us that when deriving similarity using the notion of shared nodes, not all nodes are equally important. If we want two concepts to be more similar when they have an immediate subsuming concept (e.g. *cat*[CHR:*black*] and *cat*[CHR:*brown*] because of the subsuming *cat*) than when they only share an attribute (e.g. *black* shared by *cat*[CHR:*black*] and *dog*[CHR:*black*]), we must differentiate and cannot simply define $\alpha(c)$ as a crisp set. The following is a generalization to fuzzy set based similarity [Andreasen *et al.*, 2005b], denoted as weighted shared nodes[6].

First, notice that $\alpha(c)$ can be derived as follows. Let the triple (x, y, r) be the edge of type r from concept x to concept y; let E be the set of all edges in the ontology; and let T be the top concept, which means:

$$\begin{array}{rcl} \alpha(T) & = & \{T\} \\ \alpha(c) & = & \{c\} \cup (\cup_{(c,c_i,r)\in E}\alpha(c_i)). \end{array}$$

A simple modification that generalizes $\alpha(c)$ to a fuzzy set is obtained through a function $weight(r)$ that attaches a weight to each relation type r. With this function we can generalize to:

$$\begin{array}{l} \alpha(T) = \{1/T\} \\ \alpha(c) \ = \{c\} \cup (\cup_{(c,c_i,r)\in E}\Sigma_{\mu(c_{ij})/c_{ij}\in\alpha(c_i)} weight(r) * \mu(c_{ij})/c_{ij}). \end{array}$$

$\alpha(c)$ is thus the fuzzy set of nodes reachable from concept c and modified by weights of relations $weight(r)$. As such, Andreasen et al. [2005b] define a measure of semantic similarity between two concepts as proportional to the number of nodes shared by the concepts, but where nodes are weighted according to the semantic relation by which they are reached.

For instance, from the ontology in Figure 5.2, assuming relation weights $weight(\text{ISA}) = 1$, $weight(\text{CHR}) = 0.5$ and $weight(\text{CBY}) = 0.5$, then:

> $\alpha(dog[\text{CHR}{:}black]) = 1/dog[\text{CHR}{:}black] + 1/dog + 1/animal + 0.5/black + 0.5/color + 1/anything.$

For concept similarity, the parameterized expression above can still be used applying the minimum for fuzzy intersection and the sum for fuzzy cardinality.

[6] The presentation of Weighted Shared Nodes Similarity measure in Section 5.4.4 is a rendering of the original presentation in [Andreasen *et al.*, 2005b].

$\alpha(cat[\textsc{chr}{:}black]) \cap \alpha(dog[\textsc{chr}{:}black]) = 0.5/black + 0.5/color + 1/animal + 1/anything$

$|\alpha(cat[\textsc{chr}{:}black]) \cap \alpha(dog[\textsc{chr}{:}black])| = 3.0$

The weighting of edges is very important, as it generalizes the measure so that it can be tailored for different domains with different semantic relations. It also allows differentiating between the key ordering relation, ISA and the other semantic relations when calculating similarity. The *weighted shared nodes* measure complies with all the defined properties.

5.5 Similarity Evaluation

In the previous section, a number of different similarity measures were presented and the characteristics of measures based on a set of intuitive properties were discussed. This, of course, is only one out of a variety of approaches by which similarity measures can be evaluated. Besides a theoretical study, two other evaluation methods are also prevalent in the literature: comparison to human judgments of similarity and the measures' applicability in specific natural language applications. In this section, the focus is on the comparison of the measures to human similarity judgments, while the question of applicability in relation to information retrieval is covered in the next chapter (see Section 7)[7].

Studies of human synonym judgments were performed by Rubinstein and Goodenough [1965] and repeated almost three decades later by Miller and Charles [1991]. One aspect of these experiments included asking humans to judge the similarity of pairs of words, and these judgments have in turn been the basis of the comparison of the kind of similarity measures discussed in the above section. One interesting aspect of these two experiments is that the correlation between them is 0.97, despite the more than 25-year gap between them, which somehow strengthens their validity as reference material [Miller and Charles, 1991].

In their experiment, Rubinstein and Goodenough asked two groups totaling 51 subjects to perform synonymy judgments on 65 pairs of nouns. Later, when Miller and Charles [1991] repeated Rubinstein and Goodenough's original experiment, they used a subset of 30 noun pairs from the original list of 65 pairs, where ten pairs were from the high level of synonymy, ten from the middle level and ten from the low level.

The correlation was measured using the Pearson Product Moment Correlation coefficient r between series of n measurements of two random variables X and Y, written as x_i and y_i where $i = 1, 2, \ldots, n$:

[7] The presented experiment in this section is a joint work with Rasmus Knappe, and was first presented in [Knappe, 2006].

$$r = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

Approach	Correlation
Resnik	0.744
Jiang and Conrath	0.850
Lin	0.829
Hirst and St-Onge	0.744
Leacock and Chodorow	0.816

Table 5.1: *Correlation between different similarity measures and human similarity judgments from the Miller and Charles experiment*

The goal of our experiment is threefold. First, we evaluate the weighted shared nodes measure against these two experiments, similar to what has been done for most of the other measures. Second, we investigate whether human beings actually assign an increased similarity value between concept pairs sharing attributes, e.g. the pair "*forest*" and "*graveyard*" compared to the pair "$forest$[CHR:*scary*]" and "$graveyard$[CHR:*scary*]", thereby attempting to underpin the underlying assumptions for the weighted shared nodes measure. Finally, we study the correlations between the Miller and Charles experiment and a subset of the presented measures in the last section. Using Miller and Charles as a reference, the emphasis is on comparing the weighted shared nodes measure with a selection of the other measures presented.

The correlations for the measures done by Resnik, Jiang and Conrath; Lin, Hirst and St-Onge; and Leacock and Chodorow are shown in Table 5.1 [Budanitsky, 2001].

For this purpose, a *replica* was made of the Miller and Charles experiment that included 30 concept pairs and an additional ten new compound concepts pairs. For the experiment, the staff in the computer science department at Roskilde University were asked to rate the similarity of meaning between all 40 pairs.

The results of the human synonymy judgments performed by the test persons on the parts replicated from the Miller and Charles experiments, as well as the mean ratings of the Rubinstein and Goodenough and the Miller and Charles experiments, are shown in Table 5.2. The mean ratings for the *replica* and the previous experiments are reported in the rightmost three columns.

Table 5.3 shows the correlations between the *replica* and the two previous experiments. The correlations between the replica experiment and the previous experiments are fairly good, also considering that it was performed by non-native speakers, which supports the validity of the experiment.

		Replica	R & G	M & C
car	automobile	3.82	3.92	3.92
gem	jewel	3.86	3.84	3.94
journey	voyage	3.58	3.84	3.58
boy	lad	3.10	3.76	3.82
coast	shore	3.38	3.70	3.60
asylum	madhouse	2.14	3.61	3.04
magician	wizard	3.68	3.50	3.21
midday	noon	3.45	3.42	3.94
furnace	stove	2.60	3.11	3.11
food	fruit	2.87	3.08	2.69
bird	cock	2.62	3.05	2.63
bird	crane	2.08	2.97	2.63
tool	implement	1.70	2.95	3.66
brother	monk	2.38	2.82	2.74
lad	brother	1.39	1.66	2.41
crane	implement	1.26	1.68	2.37
journey	car	1.05	1.16	1.55
monk	oracle	0.90	1.10	0.91
cemetery	woodland	0.32	0.95	1.18
food	rooster	1.18	0.89	1.09
coast	hill	1.24	0.87	1.26
forest	graveyard	0.41	0.84	1.00
shore	woodland	0.81	0.63	0.90
monk	slave	0.36	0.55	0.57
coast	forest	0.70	0.42	0.85
lad	wizard	0.61	0.42	0.99
chord	smile	0.15	0.13	0.02
glass	magician	0.52	0.11	0.44
rooster	voyage	0.02	0.08	0.04
noon	string	0.02	0.08	0.04

Table 5.2: *Replica of the Rubinstein and Goodenough and the Miller and Charles experiments*

The similarity ratings for the ten compound concepts are shown in Table 5.4. These ratings underline our assumption that humans incorporate more than one path when rating similarity. A significant increase in the similarity ratings can be seen if the similarity ratings for the atomic concept pairs are compared with the compound concept pairs that share additional aspects. For instance, the mean similarity rating increases from 0.41 for the pair (forest, graveyard) to 1.58 for the pair (scary forest, scary graveyard). Likewise, this

		Correlation
Rubinstein and Goodenough	Miller and Charles	0.97
Rubinstein and Goodenough	Replica	0.93
Miller and Charles	Replica	0.95

Table 5.3: *Correlation between the three human similarity judgment experiments*

is also true if the mean similarity ratings increase for other pairs, for example: (coast, hill) and (beautiful coast, beautiful hill), (lad, wizard) and (strange lad, strange wizard), (fast car, fancy automobile) and (very fast car, very fancy automobile).

		Mean Rating
car	automobile	3.82
gem	jewel	3.86
coast	hill	1.24
forest	graveyard	0.41
lad	wizard	0.61
glittering gem	glittering jewel	3.86
scary forest	scary graveyard	1.58
beautiful coast	beautiful hill	1.36
strange lad	strange wizard	1.32
very fast car	very fancy automobile	2.08
fast car	fancy automobile	2.05
blue-and-white gem	white jewel	2.64
blue-and-white gem	blue-and-white jewel	3.59
incomplete combustion caused by lack of oxygen	brain damage caused by lack of oxygen	1.18

Table 5.4: *Mean ratings for a subset of the experiments*

The last step in our experiment was to compare the human similarity judgments from the replica study with the corresponding similarity ratings produced by means of the weighted shared nodes measure. The weighted shared nodes measures were applied using a prototype system (see Section 7.5). In order to calculate the similarity ratings, the compound concepts to the corresponding WordNet synsets were mapped manually, similar to the manner in which disambiguation was performed in the experiments by Resnik, Jiang and Conrath; Lin, Hirst and St-Onge; and Leacock and Chodorow.

The comparison is divided into two experiments, where both the ratings from the original 30 concept pairs and the ratings from the ten compound concept pairs are compared with the ratings produced by applying the weighted shared nodes approach. In order to be comparable with the symmetrical measures proposed by Resnik, Jiang and Conrath; Lin, Hirst and St-Onge; and Leacock and Chodorow, the weighted shared nodes measure is used with $\rho = 0.5$, which makes it symmetrical.

The correlation between the mean human similarity judgments from the replica study and the ratings obtained by the weighted shared nodes measure was 0.805, while the correlations between weighted shared nodes and Rubinstein and Goodenough's experiment and Miller and Charles' experiment were 0.812, and 0.807, respectively.

In the second comparison, we arbitrarily chose initial edge weights of 1.0, 0.5, 0.5 and 0.5 for ISA, CHR, CBY and WRT, respectively. The mapping of the compound concepts to the WordNet prototype ontology was done by decomposing each compound concept and thereafter manually merging the resulting subontology with WordNet.

The final choice of weights for the different semantic relations was determined empirically. We chose to group concept pairs that use the same semantic relation and then varied the edge weight for the relation type until we obtained the maximum correlation between the ratings produced by the similarity measure and those given by human relatedness judgments for concepts including that particular relation. This was done by writing a small program that varied the edge weights of the semantic relation on a scale from 0.0 to 1.0 until maximum correlation was achieved.

In this experiment, the topmost nine concept pairs only use the CHR relation, whereas the last concept pair uses three relations.

		Semantic Relation
gem[CHR:glittering]	jewel[CHR:glittering]	CHR
forest[CHR:scary]	graveyard[CHR:scary]	CHR
coast[CHR:beautiful]	hill[CHR:beautiful]	CHR
lad[CHR:strange]	wizard[CHR:strange]	CHR
car[CHR:fast[CHR:very]]	automobile[CHR:fancy[CHR:very]]	CHR
car[CHR:fast]	automobile[CHR:fancy]	CHR
gem[CHR:blue,CHR:white]	jewel[CHR:white]	CHR
gem[CHR:blue,CHR:white]	jewel[CHR:blue,CHR:white]	CHR
combustion[CHR:incomplete]	brain death	CHR
combustion[CHR:incomplete][CBY:lack[WRT:oxygen]]	brain death[CBY:lack[WRT:oxygen]]	CHR,CBY,WRT

Table 5.5: *Relations used in the different compound concept pairs*

For this experiment, the maximum correlation with the human judgments for concept pairs containing the CHR relation was achieved with an edge weight for CHR of 0.57. For the last concept pair, we varied the weights for the two other semantic relations assuming independence and obtained maximum correspondence with weights of 0.30 and 0.36 for CBY and WRT, respectively.

		Similarity Rating	Human Similarity Judgment
glittering gem	glittering jewel	1.00	0.97
scary forest	scary graveyard	0.53	0.40
beautiful coast	beautiful hill	0.60	0.34
strange lad	strange wizard	0.70	0.33
very fast car	very fancy automobile	0.65	0.52
fast car	fancy automobile	0.74	0.51
blue-and-white gem	white jewel	0.82	0.66
blue-and-white gem	blue-and-white jewel	0.80	0.90
incomplete combustion	brain death	0.15	0.02
incomplete combustion caused by lack of oxygen	brain death caused by lack of oxygen	0.25	0.31

Table 5.6: *Similarity value for the ten compound concept pairs calculated using the weighted shared nodes measure*

Table 5.6 shows two sets of ratings for the ten compound concept pairs. The first column is the ratings from applying the concept pairs to the weighted shared nodes measure, and the second column is the ratings of human similarity judgments from the replica normalized to the unit interval. The correlation between the weighted shared nodes and the human similarity judgment is 0.87, which is a fairly high correspondence. The empirical tailoring of edge weights performed here was, of course, done on a very small set of concept pairs, and we must therefore be careful in drawing conclusions. Nevertheless, the validity is supported by the replica's correlation to the experiments performed earlier by Rubinstein and Goodenough and Miller and Charles. Furthermore, the correlation between the weighted shared nodes measure and the human similarity judgments of the compound concepts, does support our hypothesis that similarity measures can benefit from including a variety of semantic relations in the similarity measure.

5.6 Summary and Discussion

The presentation of similarity measures began with a definition of a set of intuitive, qualitative properties. Throughout the presentation, these were supplemented and the full set of properties can be summarized as follows:

- Basic properties
  - Commonality
  - Difference

    - Identity

- Retrieval-specific properties
    - Generalization

- Structure-specific properties
    - Depth
    - Multiple-Paths

During the presentation, we have discussed how the different measures comply with these properties. No measures fully obey the basic properties, since most of them define either commonality or difference but not both; for instance, the *shortest path length* measure defines difference as the distance in the ontology, but does not explicitly define commonality.

Most of the measures presented here are based on the idea of the shortest path length and will therefore inherit both the advantages and disadvantages of this measure. This group of measures can be characterized as edge-counting approaches. A common feature for these measures is that that they are simple and therefore easy to implement into retrieval systems. Another group, also based on the idea of the shortest path length, includes the measures that use *information content* as their foundation. This group of approaches differs from the edge-counting approaches because, rather than counting edges in the shortest path, they select the maximum *information content* of the least upper bound between two concepts. The major drawback of the shortest path length and *information content* approaches is that they fail to comply with the generalization property, due to symmetry. Even though similarity measures often are expected intuitively, and in some cases were also formally required to be symmetric, it has beeen shown that this is definitely not a preferred property in an information retrieval context. The *weighted shortest path* and the *generalized weighted shortest path* measures are examples of shortest path measures which solve the symmetry problem by introducing weighted edges.

Another group of measures includes the node-counting approaches, *shared nodes* and *weighted shared nodes*, which comply with all the defined properties. Like the edge-counting approaches, they are simple and can likewise be used in visualizations in a simple and straight-forward manner. The downside is that the node-counting approaches are computationally more complex than the edge-counting approaches because they include all possible paths between concepts.

The *medium-strong relation* and the generalized weighted shortest path approaches are examples of measures based on the shortest path length idea that takes all semantic relations into consideration. However, they still do not include more than one path, *the best path*, in their similarity measure, and will therefore not comply with the multiple-path property. The only measures that fully comply with the multiple-path property are the node-counting approaches.

Finally, some of the approaches share the feature of being corpus-based. They incorporate corpus analyses as an additional, and qualitatively different knowledge source for measuring the similarity between concepts in the ontology. They can be described as *integrated* due to the way that they merge knowledge about concepts (the ontology) and corpus statistics. Among the measures presented, two different approaches are used for incorporating the corpus; the measures based on Resnik's *information content* [1995; 1999] and the measures which use *instantiated ontologies* [Andreasen *et al.*, 2005a]. *Information content* has its basis in traditional corpus analysis in which semantically tagged corpora are used to measure the probability of encountering instances of concepts in the corpus. *Instantiated ontologies*, on the other hand, are built by restricting a general ontology to the (instantiated) concepts found in a given collection, which constitute two different methods for merging ontologies and corpora. The *information content* approach is based on the instantiated concepts as well as their corpus statistics, while *instantiated ontologies* only use the former. Thus both approaches identify concepts used, while "instance counting" is only performed by the *information content* approach.

Nonetheless, one of the major interesting contributions of *instantiated ontologies* is the expansion of the ontology with compound concepts extracted from a document collection. In order to benefit from this kind of information the similarity measure has to comply fully with the multiple-path property, which is only the case for the node-counting approaches.

Contrary to these integrated approaches, measures are purely based on the knowledge captured in the ontology, e.g. the *shortest path length*, and measures purely based on corpus analysis. In the above, we did not present any purely corpus-based approaches. One approach in this regard could be a "semantic network" based on the co-occurrence of senses in a sense-tagged corpus. The idea is that if two senses are co-occurring they are related. The frequency of documents in which they occur together can then be used to measure the similarity between the senses, thus only the use of the senses in a corpus contribute to the similarity.

Corpus-based measures (integrated or not) reflect a specific domain and the use of the language in this domain. As a result, information on concepts found in corpora modifies the similarity measures. In the case of *information content* one key question is whether or not it is appropriate to generalize from one corpus to another and whether domains can be shifted. This would naturally primarily depend on the generality of the corpus source. The *information content* approach by Resnik [1995; 1999], in which WordNet and the

Brown Corpus are used as the foundation for the similarity measure, constitutes a generalized similarity model (see Section 5.3.1). The results from the comparison between the *information content* similarity measures and human similarity judgments underpin this conclusion (see Section 5.5), thus indicating that it is appropriate to generalize similarity measures from relatively small resources due to the fact that the *information content* similarity does not perform significantly differently from the other measures in the text.

The main goal in the development of the node-counting similarity methods is to include as many aspects as possible when estimating similarity, and hence consider more than one "path" between the concepts being compared. The *shared nodes* approach and the *weighted shared nodes* approach both consider what could be defined as multiple paths, since similarity is dependent on the number of shared nodes on all upward paths from the concepts to the top. The number of shared nodes between two concepts expresses their commonality, and the more nodes that are shared, the higher the similarity. However, as described earlier, not all nodes on paths connecting concepts contribute equally to the definition of concepts.

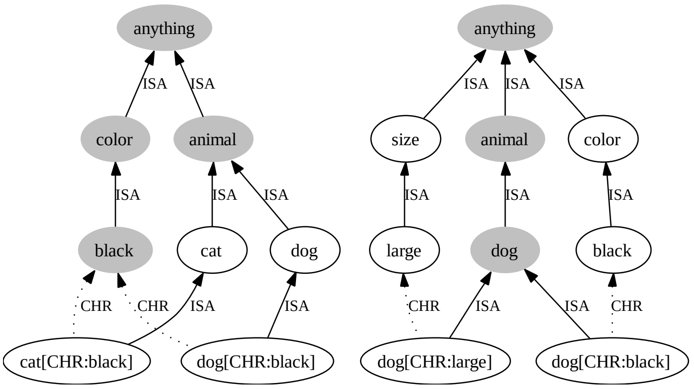


Figure 5.10: *Two example ontologies illustrating that not all shared nodes are equally important*

Consider the example in Figure 5.10, where the similarity between the concepts *dog*[CHR:*black*] and *cat*[CHR:*black*] would be greater than the similarity between *dog*[CHR:*black*] and *dog*[CHR:*large*] according to the *shared nodes* approach. Thus the first two concepts share four nodes, whereas the last two concepts share only three nodes. This is counter intuitive since the concept inclusion relation (ISA) should have higher importance than the "characterized-by" relation (CHR). This problem is the motivation for the *weighted shared nodes* approach, where the attachment of weights to relations is the remedy. One benefit of this transformation is that the *weighted shared nodes* measure can be tailored to include a wide number of aspects when calculating similarity through the weights, including important properties like commonality, difference, generalization, specialization and multiple paths.

The main purpose of the experiments performed in this chapter was to evaluate the assumption behind the *shared nodes* measures, namely that a similarity measure benefits from considering more than just the concept inclusion relation when measuring similarity between compound concepts. The evaluation was done by comparing the similarity ratings from the *weighted shared nodes* measure with the average similarity ratings produced by humans for a set of concept pairs. The human similarity judgment experiment was conducted as a replica of previous experiments performed by Rubistein and Goodenough and Miller and Charles and we obtained correlations of 0.93 and 0.95 between our human replica study and the two previous studies. The majority of similarity measures presented in this chapter cannot include other aspects then those denoted by the concept inclusion relation, therefore the first experiment considered only the concept inclusion relation. We obtained correlations of 0.805, 0.812 and 0.807 for our replica, the Rubinstein and Goodenough experiment, and the Millar and Charles experiment, respectively. These are higher than the correlations obtained by Resnik and Hirst and St-Onge, but slightly lower than the correlations obtained by Lin, Jiang and Conrath and Leacock and Chodorow.

Because these original experiments only measured similarity between atomic concepts, we expanded the set of concept pairs to include ten additional compound concept pairs. The high correlation between the mean similarity ratings assigned for the original 30 concept pairs in our replica study are seen as an indication of the validity of the values assigned by the same test persons for the compound concept pairs.

Our experiment with compound concepts showed that humans assign increased similarity for concepts sharing aspects and attributes, and that the *weighted shared nodes* measure assigns values that correlate highly with this. A very high correlation of 0.87 was obtained between our replica experiment for the ten compound concept pairs and the similarity values assigned by the *weighted shared nodes.* We consider this to indicate that the inclusion of additional aspects in the form of other semantic relations corresponds to a large extent to how humans rate similarity, which in turn validates not only the assumption behind, but also the actual properties of the *weighted shared nodes* measure.

The aim of studying ontological similarity measures in this thesis is to improve the retrieval of information. Whether or not the *weighted shared nodes* measure would contribute to this purpose is not a conclusion that can be drawn directly from this chapter. We can only conclude that if we were able to incorporate the similarity ratings calculated using the proposed measure into our evaluation methodology, then we could obtain the means for performing ontology-based query expansion, which to some extent follows the way humans compare and relate concepts as well as the structure and relations of the ontology.

Finally, as a more general observation, we would like to emphasize that the modeling of similarity or relatedness functions is far from objective. It is not possible to define optimal functions either in general or in a domain specific case. We can only attempt to make flexible and parameterized functions on the basis of obvious "intrinsic" properties with intuitive interpretations and then adjust and evaluate these functions on an empirical basis. The tailoring of the proposed similarity function can be done according to specific needs regarding the structure and relations of the ontology covering the domain.

Chapter 6

Query Evaluation

The purpose of this chapter is to fuse together the ideas of ontological indexing and ontological similarity into a realistic information retrieval scenario, and in so doing, promote semantics in the document retrieval process.

The descriptions revealed by ontological indexing can vary in their degree of abstraction from simple atomic concepts to complex Ontolog expressions. The query evaluation should naturally adapt to the type of description in order to obtain the full power of the ontological indexing. To achieve this, a generalized fuzzy retrieval model scalable to different kinds of description representations is used.

In the utilization of ontological similarity in the query evaluation process, similarity measures compare concepts, while the query evaluation compares descriptions, i.e. descriptions of queries are compared to descriptions of documents. The most obvious solution for introducing ontological similarity in the query evaluation is to modify the descriptions to include revealed knowledge, more specifically, to expand the descriptions with similar concepts found using ontological similarity. An approach we have named "semantic expansion".

Moreover, the introduction of ontological knowledge in the retrieval process appears to have two different obvious prospects. One concerns the ontology as a target for querying and hence the retrieval of knowledge instead of documents, while the other one involves the use of the structure of the ontology for surveying, navigating, and visualizing the domain covered by the system's document base.

In conventional information retrieval, the evaluation is measured by recall and precision, the ratio of the relevant documents found and all possible relevant documents, as well as the ratio of relevant and non-relevant documents in a given retrieval (see Section 2.2). Obviously this evaluation criterion should apply for ontology-based retrieval systems, too. The testing process of query evaluation should also include user tests, since a measure of the retrieval evaluation would be only one view of the retrieval process in general. Here we make suggestions to indicate improvements and justify the intentions of this approach using superficial experiments, allowing room for further work using in-depth and real life experiments.

6.1 Semantic Expansion

The purpose of the expansion of descriptions is to reduce the gap between user intention and system interpretations of queries and documents. An expansion that adds relevant aspects to a description will lead to increased recall and the assumption is that this, in turn, will lead to better answers because the resulting reduced precision can be dealt with through inherent modification mechanisms of the evaluation principle, for instance, the ranking of results. The *semantic expansion* introduced here can be seen as a method for softening the interpretation of a concept, which is given by the position of the concept in the ontology. To achieve semantic expansion, a set of similar concepts defined by some kind of ontology-based similarity measure is used.

In an ontology-based system, the idea is to map information found in documents or queries into an ontology, and in so doing, draw closer to the meaning of the information. Normally, some kind of sense disambiguation is used to determine the exact position in the ontology, which is a process of selecting the right interpretation among the possible ones (see Section 4.2.2). This selection would of course exclude some interpretations, which due to the uncertainty, could actually be the right ones. One solution to this problem is not to choose, but instead weigh different interpretations proportionally in relation to how likely they are in the given context. Irrespectively of whether interpretations are disambiguated or weighted, a semantic expansion can be used to reveal similar information in the query evaluation process.

The aspects considered in determining the strategy for a semantic expansion are the "cost" in terms of time and space, and the degree of flexibility; that is, the possibility of managing the constraints that control the expansion process. Examples of such constraints are thresholds determining the size of the expansion and the degree of similarity of the expanded concepts. However, the choice of similarity function and thresholds concerning this function are also considered as constraints of the expansion. Parts of the expansion process may be computationally complex and may therefore be essential to handle prior to the query evaluation in order to reduce the response time of the retrieval. Consequently, this can lead to a reduction in flexibility, as some constraints may have to be fixed do to this preprocessing.

The expansion of documents, denoted *object expansion*, is a process where concepts found in the descriptions of documents are expanded. Object expansion is normally done as a part of the indexing process, and not during the query evaluation. The advantage of this type of expansion is that the process can be moved from the query evaluation to a preprocessing (normally, the indexing process). This can reduce the amount of time consumed at query time, given that the time spent on searching and fetching data in the considerably larger (expanded) index does not counterbalance the time gained by the preprocessing. The disadvantages of object expansion are that it can be very space consuming and can reduce flexibility, since a possible consequence of preprocessing the expansion is that some of the constraints become fixed.

Another kind of expansion is *query expansion*, where the description of the given query is expanded. The "cost" of query expansion is normally trifling, due to the small size of queries, in contrast to object expansion, and therefore the processing of queries can easily be done at query time. One major benefit of query expansion is the flexibility, since all the constraints are unbound. The drawback is that queries often are very limited (short), and therefore difficult to interpret, which can influence the quality of the expansion.

In this thesis, query expansion is the chosen expansion strategy because full access to all constraints is necessary in order to experiment with different models and parameters.

6.1.1 Concept Expansion

The goal of a *concept expansion* is to expand the interpretation of a given concept with closely related concepts in order to achieve a match on "conceptual content" rather than on specific words or concepts, as well as to compensate for the uncertainty due to ambiguous senses in natural language. Concepts are closely related when they have a high degree of similarity, which for the similarity measures in use here means concepts that are positioned closely together in the ontology with respect to distance. The quality of a concept expansion could be thought of as the ability of the expansion to substantiate the user's interpretation of a given concept. In the information retrieval system, this would equal the ability of the system to reveal documents judged relevant by the users. To establish whether a given expansion is supportive or deceptive is difficult since similarity is rather subjective and most likely needs to be determined empirically for a given system or left open to the users by giving options to justify the parameters of the expansion.

Let us assume that the foundation of the expansion is an instantiated ontology describing the domain of the given retrieval system; hence, we have a finite ontology. In Figure 6.1, a simple instantiated ontology of the concepts *cat*[CHR: *black*] and *poodle*[CHR: *black*] is shown. This ontology is used as the foundation for the examples given in this section.

The similarity functions discussed in Chapter 5 compare pairs of concepts and calculates their degree of similarity. The semantic expansion requires a function, which for a given concept returns a fuzzy set of similar concepts

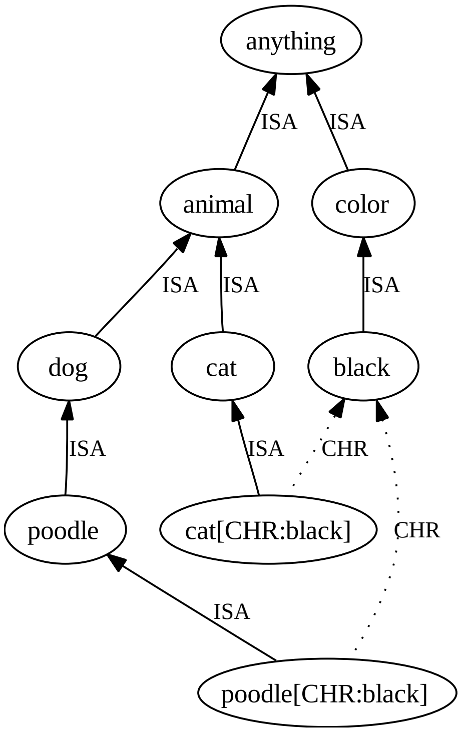


Figure 6.1: *An instantiated ontology over the concepts **cat**[**chr**: **black**] and **poodle**[**chr**: **black**]*

where the membership function defines the similarity measure. A simple example is a function that expands to all related concepts

$$similar(c) = \sum_{c_i \in C} sim(c, c_i)/c \tag{6.1}$$

assuming an instantiated ontology where $C = \{c_1, c_2, \ldots, c_n\}$ is the set of all concepts, and where the similarity function *sim*, for instance, could be defined as the shared note similarity:

$$sim(x, y) = \rho \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1 - \rho) \frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|} \tag{6.2}$$

Query expansion can be either static or dynamic. Static means the computation of the similarity is done in advance, and dynamic means that the computation is done at the time of the query evaluation.

A simple static approach is to represent the degree of similarity for any pair of concepts in the ontology in a *similarity matrix*. The advantage of this approach is that the expansion easily can be derived from the similarity matrix without the need of traversing the ontology. The disadvantages of a similarity matrix are the lack of flexibility, as the similarity measure is fixed, the potential computational complexity and the rather space consuming attributes of large ontologies.

The space consumption can be reduced by using a threshold for the smallest degree of similarity to be included in the similarity matrix, thereby changing the similarity matrix into a sparse matrix. This will, however, not reduce the cost of computing similarities between concepts, and a similarity matrix would also need recalculation whenever the ontology changes.

The flexibility can be partially regained using *partial preprocessing*, an approach where partial results used for the computation of the similarity measure are stored. Using the similarity function (6.2) as an example, the computational complexity is bound to the upwards expansion $\alpha(x)$ of the concepts and the intersection between these upwards expansion for pairs of concepts. One example of partial preprocessing which maintains some of the flexibility is a matrix with information about the cardinality of shared nodes for any pair of concepts in the ontology. Table 6.1 shows the matrix for the cardinality of the shared nodes as the value of the (i, j)'th entry of the ontology in Figure 6.1.

	anything	*animal*	*color*	*dog*	*cat*	*black*	*poodle*	*cat*[CHR:*black*]	*poodle*[CHR:*black*]
anything	1	1	1	1	1	1	1	1	1
animal	1	2	1	2	2	1	2	2	2
color	1	1	2	1	1	2	1	2	2
dog	1	2	1	3	2	1	3	2	3
cat	1	2	1	2	3	1	2	3	2
black	1	1	2	1	1	3	1	3	3
poodle	1	2	1	3	2	1	4	2	4
cat[CHR:*black*]	1	2	2	2	3	3	2	6	4
poodle[CHR:*black*]	1	2	2	3	2	3	4	4	7

Table 6.1: *The cardinality of the shared nodes of concepts in the ontology from Figure 6.1*

A dynamic expansion approach, on the other hand, necessitates time optimizations as the expansion is computed at query time. The similarity measure is influenced by distance in the ontology and most closely related concepts are therefore the concepts in the "ontological surroundings" of the concept that we want to expand. One obvious solution is to reduce the traversing of the ontology to only the set of closely positioned concepts we want to include in the expansion. The idea is that an expansion function starts traversing the ontology at the initial concept (the concept we want to expand) and follows the paths through its relations. This idea is similar to a technique called *spreading-activation* [Collins and Loftus, 1988] used in semantic networks to find the least upper bound between two or more nodes in semantic networks. In spreading-activation, nodes are activated when the activation traverses the network, which means the traversing can be controlled in order not to visit the same node twice. An expansion function inspired by this technique should then start by activating the initial concept, followed by all concepts directly related in the ontology, followed by their related concepts, and so forth, as activation spreads across relations in the ontology. Ordinary spreading-activation can quickly reach every concept in the ontology, and since many of the concepts found this way would not be pertinent, heuristics must be introduced to constrain the search algorithm in order to favor the the concepts that are most similar. The three most obvious parameters for such constraints are the degree of similarity, the number of edges from the initial concept (distance), and the number of concepts in the expansion (cardinality). Only the first parameter guarantees that all the concepts found by the expansion are inside a given limit with respect to the degree of similarity, since concepts within a given number of edges from the initial concept or the cardinality of the expansion can not ensure this. Combinations of the constraints are naturally possible, for instance, when a given degree of similarity in combination with a given maximal cardinality of the set of expanded concepts, which can be used to compensate for varying density in the ontology, thus restraining the expansion whenever the density is very high in order not to generate large sets of expansions.

Let C be the set of concepts and the similarity function *sim* (6.2) the shared note similarity, then the expansion can be defined as:

$$expansion_{\alpha}(c) = \{sim(c, c')/c | sim(c, c') \geq \alpha\} \tag{6.3}$$

where $c, c' \in C$ and α a threshold for the degree of similarity.
Take, for example, the ontology in Figure 6.1 and the expansion function (6.3) with (6.2) as the *sim* function where $\rho = 0.8$, then the expansion of the concept *cat* is:

$$\begin{array}{rl} & 0.73/animal + 0.90/cat[\textsc{chr}{:}black]+ \\ expansion_{spreading}(cat) = & 0.47/anything + 0.67/dog + 0.33/black+ \\ & 0.37/color + 0.63/poodle + 0.59/poodle[\textsc{chr}{:}black] \end{array}$$

where each line in the fuzzy set refers to an activation step in the expansion.

6.2 Query Evaluation Framework

In the indexing process, the documents are analyzed and the concepts are extracted and mapped into the ontologies. The result of the indexing process is reflected in an instantiated ontology describing the domain of the documents stored in the retrieval system. The document indexing combined with the instantiated ontology forms the basis for the query evaluation, and the input to the query evaluation is queries which may or may not be semantically expanded.

In principle, any retrieval model that supports weighted descriptions for both queries and documents is usable for the query evaluation framework introduced here. Thus, of the models introduced in Chapter 2, both the vector and fuzzy model fulfill this requirement. However, the fuzzy retrieval model has advantages over the vector model for the purpose in this context, especially with respect to the opportunity of modifying the logical interpretation of queries and the grouping of information in the queries, which is important in order to grasp the intrinsic structure of the natural language obtained by natural language processing.

6.2.1 Simple Fuzzy Query Evaluation

The foundation of fuzzy information retrieval is introduced in Section 2.1.5. Descriptions of both queries and documents are fuzzy sets where membership functions describe the strength of the relation between descriptors and descriptions. In the simplest case, a descriptor can either be part of a description or not, while evaluation in the model would in this case resemble "best match" in the Boolean model.

The general idea in simple fuzzy query evaluation is to compare the description of a query represented by a fuzzy set to the fuzzy sets that define the descriptions of documents in order to find the documents that best match the query. The result of a query evaluation is a fuzzy subset of the documents which match the query, where the retrieval status value (RSV) is defined by the membership function $\mu_Q(d)$ as the degree of relevance of document d to query Q.

One way to define this membership function is by use of the relative sigma count:

$$RSV_Q(d) = \mu_Q(d) = \frac{\sum_{c \in Q} \min(\mu_d(c), \mu_Q(c))}{\sum_{c \in Q} \mu_Q(c)}$$

where $\mu_d(c)$ is a membership function which defines the strength of the relation between a given concept c and a document d and $\mu_Q(c)$, a membership function that defines the strength of the relation between a given concept c and the query Q.

The indexing of documents is defined as a binary fuzzy index relation, which defines the relation between documents and concepts (see 2.1.5 for the definition). As an example of simple fuzzy query evaluation consider a retrieval system that consists of the following set of documents, D, with three documents and the set of concepts, C, with three concepts:

$$\begin{aligned} D &= \{d_1, d_2, d_3\} \\ C &= \{c_1, c_2, c_3\}. \end{aligned}$$

The indexing would then define the degree to which each concept c_i is assigned to each document d_j:

$$\begin{aligned} I = \quad & \{1/(d_1, c_1) + 1/(d_1, c_2) + 0/(d_1, c_3) + \\ & 1/(d_2, c_1) + 0/(d_2, c_2) + 1/(d_2, c_3) + \\ & 0/(d_3, c_1) + 1/(d_3, c_2) + 1/(d_3, c_3)\}, \end{aligned}$$

and the descriptions of specific documents can be derived from the binary fuzzy index relation I as:

$$\begin{aligned} I_{d_1} &= \{1/c_1 + 1/c_2 + 0/c_3\} \\ I_{d_2} &= \{1/c_1 + 0/c_2 + 1/c_3\} \\ I_{d_2} &= \{0/c_1 + 1/c_2 + 1/c_3\}. \end{aligned}$$

In a query evaluation with the query $Q = \{c_1, c_2\}$, a relative sigma count of $\mu_Q(d)$, and the above example system, the evaluation would be computed as follows:

$$\mu_Q(d_1) = \frac{\min(\mu_{d_1}(c_1), \mu_Q(c_1)) + \min(\mu_{d_1}(c_2), \mu_Q(c_2))}{\mu_Q(c_1) + \mu_Q(c_1)} = \frac{\min(1,1) + \min(1,1)}{1+1} = 1$$

$$\mu_Q(d_2) = \frac{\min(\mu_{d_2}(c_1), \mu_Q(c_1)) + \min(\mu_{d_2}(c_2), \mu_Q(c_2))}{\mu_Q(c_1) + \mu_Q(c_1)} = \frac{\min(1,1) + \min(0,1)}{1+1} = 0.5$$

$$\mu_Q(d_3) = \frac{\min(\mu_{d_3}(c_1), \mu_Q(c_1)) + \min(\mu_{d_3}(c_2), \mu_Q(c_2))}{\mu_Q(c_1) + \mu_Q(c_1)} = \frac{\min(0,1) + \min(0,1)}{1+1} = 0.$$

The result of a given query Q can then be ordered by the value of $\mu_Q(d)$ (the RSV) with the best match first, as in the above example. All non-relevant documents can be excluded from the resulting fuzzy set by use of a threshold for the lowest acceptable value of RSV. For instance, $RSV > 0$, which would exclude document d_3 from the result in the above example. This example also shows that in the general case where all concepts in C are assigned to every document in D by the binary fuzzy index relation, every document has to be evaluated. Obviously, some kind of optimization is preferable since the subset of documents relevant for average queries is far from the complete set of documents. One such optimization is to define a subset P_Q of documents

D with the set of documents that can possibly satisfy the query Q. A set of document descriptions indexed by a concept x, a *document list*, can be defined on the basis of the binary fuzzy indexing relation I, given $x \in Q$, as the fuzzy subset I_x of document descriptions about x:

$$I_x = \{\mu_{I_x}(d)/(d) | d \in D; \mu_{I_x}(d) = \mu_I(d, x)\}$$

The set P_Q of document descriptions that possibly satisfy the query is then given by:

$$P_Q = \bigcup_{x \in Q} I_x$$

where P_Q represents the union of document lists associated with each of the descriptors in the query. Since the objective is to locate only the top set of ranking document descriptions, we can define:

$$RSV_{Q(\alpha)} = \{\mu_Q(d)/d | \mu_Q \geq \alpha; d \in P_Q\}$$

as the documents which best fulfill the query, which is restricted to the document descriptions in P_Q, instead of the full collection D and the subset of documents with $RSV_{Q(\alpha)}$ greater or equal to the threshold α.

6.2.2 Ordered Weighted Averaging Aggregation

In the above simple fuzzy query evaluation, queries are evaluated by the relation between the fuzzy set describing the query and the fuzzy set describing documents (in the above example as the relative sigma count between these to fuzzy sets). One way to generalize this retrieval model is to introduce order weighted averaging (see Section 2.1.5)[1]. In order to support this aggregation, the descriptors in queries have to be evaluated separately, that is, instead of a membership function for the query $\mu_Q(d)$, we need to define membership functions for each descriptor in the query, $\mu_{q_i}(d)$, such that the value $\mu_{q_i}(d) \in [0, 1]$ is the degree to which the document description d satisfies the query descriptor q_i. The overall valuation of d is thus:

$$\mu_Q(d) = OWA_W(\mu_{q_1}(d), \ldots, \mu_{q_n}(d)).$$

where W is the weighting vector used in the aggregation (see Section 2.1.5 for a description of aggregation with weighting vectors).

The ordered weighted averaging operator aggregation principle is very flexible and may further include importance weighting in the form of an n-vector $M = (m_1, \ldots, m_n)$, $m_j \in [0, 1]$ giving attribute importance to $q_1, \ldots, q_n$ such

[1] The presentation of Ordered Weighted Averaging Aggregation in Section 6.2.2 is a rendering of the original presentation given in [Andreasen *et al.*, 2005b].

that, for instance, $M = (1, 0.8, 0.8, \ldots)$ gives more importance to q_1, while importances are not discriminated with $M = (1, 1, \ldots)$. The introduction of attribute importance corresponds to a modification of the valuation $\mu_Q(d)$ into $OWA_W(\mu_{q_1}(d) * m_1, \ldots, \mu_{q_n}(d) * m_n)$.

Recall that the aggregation may be modeled by a "linguistic quantifier", which basically is an increasing function $K : [0, 1] \rightarrow [0, 1]$ where $K(0) = 0$ and $K(1) = 1$, such that the order weights are prescribed as:

$$w_j = K(\frac{j}{n}) - K(\frac{j-1}{n}).$$

Linguistic quantifiers can lead to values of W and we can model, for instance, a quantifier $EXISTS$ by $K(x) = 1$ for $x > 0$, FOR-ALL by $K(x) = 0$ for $x < 1$, and SOME by $K(x) = x$, while one possibility (of many) to introduce $MOST$ is by a power of $SOME$, e.g. $K(x) = x^3$. Thus, we have a general query expression:

$$Q =< q_1, \ldots, q_n : M : K >$$

where $q_1, \ldots, q_n$ are the query descriptors, M specifies the importance of weighting for these, and K specifies a linguistic quantifier and thereby indicates an order weighting. The corresponding generalized valuation function is:

$$\mu_Q(d) = OWA_{M,w(K)}(\mu_{q_1}(d), \ldots, \mu_{q_n}(d)) \tag{6.4}$$

assuming a function $w(K) \rightarrow [0, 1]^n$ that maps onto the set of order-weights corresponding to quantifier K.

6.2.3 Hierarchical Aggregation

A hierarchical approach to aggregation generalizing the ordered weighted averaging operator is introduced in [Yager, 2000]. Basically, hierarchical aggregation extends ordered weighted averaging to capture nested expressions[2].

Query attributes may be grouped for individual aggregation and the language is orthogonal in the sense that aggregated values may appear as arguments to aggregations. Thus, queries may be viewed as hierarchies. For example, the following nested query expression can be posed:

1. $< \mu_{q_1}(d),$
2. $\quad < \mu_{q_2}(d), \mu_{q_3}(d),$
3. $\qquad < \mu_{q_4}(d), \mu_{q_5}(d), \mu_{q_6}(d) : M_3 : K_3 >$
4. $\qquad : M_2 : K_2 >,$

[2] The presentation of Hierarchical Aggregation in Section 6.2.3 and 6.2.4 is a minor modified rendering of the original presentation given in [Andreasen *et al.*, 2005b].

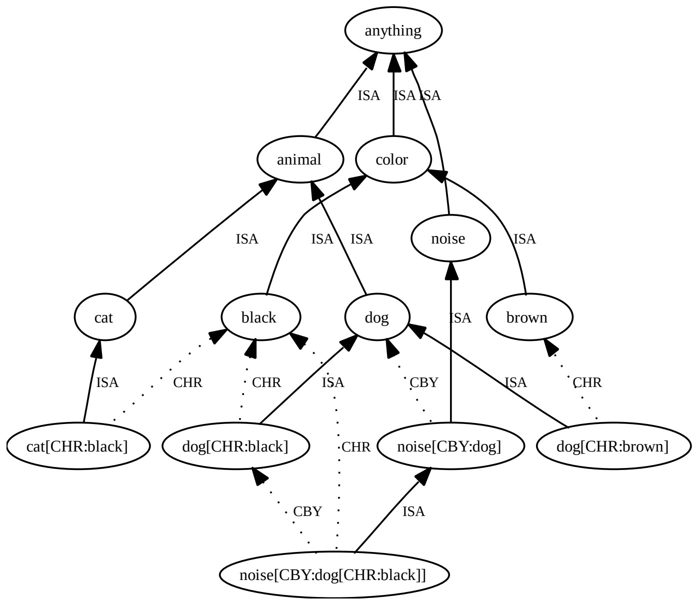


Figure 6.2: *A simple instantiated ontology based on the set of instantiated concepts cat[CHR:*black*], dog[CHR:*black*], dog[CHR:*brown*], $noise$[CBY:dog[CHR:*black*]]*

5. $: M_1 : K_1 >$

Again, $\mu_{q_i}(d)$ measures the degree to which attribute q_i conforms to document description d, while M_j and K_j are the importance and quantifier applied in the j'th aggregate. In the expression above, $M_1 : K_1$ parameterizes aggregation at the outermost level of the two components $\mu_{q_1}(d)$ and the expression in lines 2 to 4. $M_2 : K_2$ parameterizes the aggregation of the three components $\mu_{q_2}(d)$, $\mu_{q_3}(d)$, and the innermost expression (line 3), while $M_3 : K_3$ parameterizes aggregation of the three components $\mu_{q_4}(d)$, $\mu_{q_5}(d)$, and $\mu_{q_6}(d)$.

6.2.4 Hierarchical Query Evaluation Approaches

Two major cases of description structure are distinguished between – simple unested sets and nested sets.

Aggregation on unnested descriptions

The simple set-of-descriptors structure for descriptions admits a straightforward valuation approach for a similarity query description:

$$Q_{sim} =< q_1, \ldots, q_n : (1, 1, \ldots) : SOME > \tag{6.5}$$

The aggregation here is simple in that importance is not distinguished and SOME, corresponding to the simple average, is used as a quantifier. A valuation can be:

$$\mu_{Q_{sim}}(d) = F_{(1,1,\ldots),w(SOME)}(\mu_{q_1}(d), \ldots, \mu_{q_n}(d))$$

with the individual query-descriptor valuation functions as:

$$\mu_{q_i}(d) = maximum_j\{x|x/d_j \in similar(q_i)\}$$

For instance, the query $Q =< dog[\text{CHR}:black], noise >$ and the instantiated ontology in Figure 6.2 with a threshold on 0.4 means that:

$similar(dog[\text{CHR}:black]) =$
$\quad 1/dog[\text{CHR}:black] + 0,7/dog[\text{CHR}:brown]+$
$\quad 0,68/dog + 0,6/cat[\text{CHR}:black]+$
$\quad 0,58/noise[\text{CBY}:dog[\text{CHR}:black]] + 0,52/animal+$
$\quad 0,45/cat + 0,45/black + 0,42/noise[\text{CBY}:dog]$

$similar(noise) =$
$\quad 1,00/noise + 0,90/noise[\text{CBY}:dog]+$
$\quad 0,87/noise[\text{CBY}:dog[\text{CHR}:black]]+$
$\quad 0,60/anything + 0,50/animal + 0,50/color+$
$\quad 0,47/cat + 0,47/black + 0,47/dog + 0,47/brown+$
$\quad 0,44/cat[\text{CHR}:black] + 0,44/dog[\text{CHR}:black]+$
$\quad 0,44/dog[\text{CHR}:brown]$

and, for instance, the following:

$\mu_{Q_{sim}}(\{noise[\text{CBY}:dog]\}) = 0.66$
$\mu_{Q_{sim}}(\{noise[\text{CBY}:dog[\text{CHR}:black]]\}) = 0.73$
$\mu_{Q_{sim}}(\{dog, noise\}) = 0.84$
$\mu_{Q_{sim}}(\{cat[\text{CHR}:black], noise\}) = 0.80$

Finally, when considering the example of $Q =< noise[\text{CBY}:dog[\text{CHR}:black]] >$ and the instantiated ontology in Figure 6.2 with a threshold of 0.4, the result is:

$similar(noise[\text{CBY}:dog[\text{CHR}:black]]) =$
$\quad 1,00/noise[\text{CBY}:dog[\text{CHR}:black]]+$
$\quad 0,73/noise[\text{CBY}:dog] + 0,52/dog[\text{CHR}:black]+$
$\quad 0,47/noise + 0,40/dog + 0,40/black$

and among valuations are the following:

$\mu_{Q_{sim}}(\{noise[\text{CBY}:dog[\text{CHR}:black]]\}) = 1.00$
$\mu_{Q_{sim}}(\{noise[\text{CBY}:dog], black\}) = 0.73$
$\mu_{Q_{sim}}(\{noise, dog[\text{CHR}:black]\}) = 0.52$
$\mu_{Q_{sim}}(\{noise, dog, black\}) = 0.47$

Nested aggregation on unnested descriptions

An alternative is to expand the query description Q to a nested expression:

$$\begin{aligned}
&\mu_{Q_{sim}}(d) = \\
&\quad << \mu_{q_{11}}(d), \ldots, \mu_{q_{1k_1}}(d) : M_1 : K_1 >, \\
&\quad\ \ < \mu_{q_{21}}(d), \ldots, \mu_{q_{2k_2}}(d) : M_2 : K_2 >, \\
&\quad\quad\ \ldots, \\
&\quad\ \ < \mu_{q_{n1}}(d), \ldots, \mu_{q_{nk_n}}(d) : M_n : K_n >, \\
&\quad : M_0 : K_0 >
\end{aligned}$$

where for each q_i, we set:

$$< \mu_{q_{i1}}/q_{i1}, \ldots, \mu_{q_{ik_i}}/q_{ik_i} >= similar(q_i)$$

and use as individual valuation:

$$\mu_{q_{ij}}(d) = \begin{cases} \mu_{q_{ij}}, & \text{when } q_{ij} \in \{d_1, \ldots, d_m\} \\ 0, & \text{otherwise} \end{cases}$$

In the case that we use equal importance and the following combination of quantifiers:

$$\begin{aligned}
&\mu_{Q_{sim}}(d) = \\
&\quad << \mu_{q_{11}}(d), \ldots, \mu_{q_{1k_1}}(d) : (1, 1, \ldots) : EXIST >, \\
&\quad\ \ < \mu_{q_{21}}(d), \ldots, \mu_{q_{2k_2}}(d) : (1, 1, \ldots) : EXIST >, \\
&\quad\quad\ \ldots, \\
&\quad\ \ < \mu_{q_{n1}}(d), \ldots, \mu_{q_{nk_n}}(d) : (1, 1, \ldots) : EXIST >, \\
&\quad : (1, 1, \ldots) : SOME >
\end{aligned}$$

we get a valuation identical to that of the function in the previous subsection. However, with the nested expression, the option also exists to use an unweighted similarity function and to introduce the differentiation of influence from relations by importance weighting, as indicated below. For the query description $Q =< dog[\text{CHR}{:}black], noise >$:

$$\begin{aligned}
&\mu_{Q_{sim}}(d) = \\
&\quad << \mu_{q_{dog[\text{CHR}:black]}}(d), \mu_{q_{dog}}(d), \mu_{q_{black}}(d), \ldots \\
&\quad\quad\quad\quad\quad\quad : (1, 1, 0.5, \ldots) : EXIST >, \\
&\quad\ \ < \mu_{q_{noise}}(d), \ldots : (1, 1, \ldots) : EXIST > \\
&\quad : (1, 1, \ldots) : SOME >
\end{aligned}$$

where the importance of, for instance, the different elements of noun phrases are stressed, e.g. that *dog*[CHR:*black*] (the noun phrase) and *dog* (the head of the noun phrase and the generalization) both have importance 1, while *black* (the modifier) only has importance 0.5.

Aggregation on nested descriptions

In some cases, when text is processed by partial analysis, as described in Chapter 4, an intrinsic structure appears as the most obvious choice for the description. A multi-level parsing, for instance, a two-phase parser that groups words in the documents into groups corresponding to noun phrases in the first phase, and derives compound descriptors from the words in each noun phrase individually in the second, is an example of a parsing that creates an intrinsic structure. Thus, we have as an intrinsic structure from the first phase – a set of sets (or lists) of words. Now, if it was always possible to extract a unique compound concept as a descriptor from an inner set, the resulting intrinsic structure from the second phase would be the single set, as assumed above. However, in many cases, it is not possible, and the information is thus lost due to flattening to a single set. This indicates that a set-of-sets structure is a better description structure and suggests that descriptions are sets of sets of descriptors such that the query structure is:

$$Q =< Q_1, \ldots, Q_n >=<< q_{11}, \ldots, q_{1k_1} >, \ldots, < q_{n1}, \ldots, q_{nk_n} >>$$

where the Q_i's are sets of descriptors $q_{ij}, j = 1, \ldots, k_i$, and a text index is:

$$d = \{d_1, \ldots, d_m\} = \{\{d_{11}, \ldots, d_{1l_1}\}, \ldots, \{d_{m1}, \ldots, d_{ml_m}\}\}$$

where the d_i's are sets of descriptors $d_{ij}, j = 1, \ldots, l_i$.

This, however, demands a modified valuation and since, in this case, the initial query expression is nested, a valuation over a nested aggregation also becomes the obvious choice. First of all, note that the grouping of descriptors in descriptions has the obvious interpretation of a closer binding of descriptors within a group than across different groups (in contrast to the simple fuzzy query evaluation, where queries are evaluated as single fuzzy sets). As a result, $q_{ij}(d)$ cannot be evaluated individually, but has to compare groups, for instance, by a restrictive quantification over $q_{i1}(d_j), \ldots, q_{ik_i}(d_j)$, as well as by using an $EXIST$ quantification over j to get the best matching d_j for a given Q_i. A valuation can thus be:

$$\begin{aligned}
&\mu_{Q_{sim}}(d) = \\
&\quad <<< \mu_{q_{11}}(d_1), \ldots, \mu_{q_{1k_1}}(d_1) : M_{11} : MOST >, \\
&\qquad\qquad \ldots, \\
&\qquad < \mu_{q_{n1}}(d_1), \ldots, \mu_{q_{nk_n}}(d_1) : M_{n1} : MOST > \\
&\quad : M_1 : EXIST >, \\
&\qquad\qquad \ldots, \\
&\quad << \mu_{q_{11}}(d_m), \ldots, \mu_{q_{1k_1}}(d_m) : M_{11} : MOST >, \\
&\qquad\qquad \ldots, \\
&\qquad < \mu_{q_{n1}}(d_m), \ldots, \mu_{q_{nk_n}}(d_m) : M_{n1} : MOST >
\end{aligned}$$

$: M_m : EXIST >,$
$: M_0 : SOME >$

The individual query-descriptor valuation functions can be set to:

$$\mu_{q_{ij}}(d_k) = maximum_l\{x|x/d_{kl} \in similar(q_{ij})\}$$

As opposed to the single set description example above, the q_{ij}s here are the original descriptors from the query. While choices of inner quantifiers are significant for correct interpretation, the choice of $SOME$ at the outer level for the component description is just one of many possible choices to reflect user preference of overall aggregation.

6.3 Knowledge Retrieval

The goal of posing queries to an information retrieval system is information, which would normally be documents covered by the system. In that case, the objective of the querying is *document retrieval*. Another objective could be *knowledge retrieval*, where the goal of querying is the knowledge covered by the indexing of the information in a retrieval system. A simple example is information about occurrences of particular words, combinations of words, etc. in statistical approaches. The introduction of ontologies makes this kind of querying even more interesting as it gives the opportunity for questioning knowledge "hidden" in the mapping between the information in the documents and the information in the ontologies[3].

An obvious extension to a framework where the evaluation of queries to documents involves the interpretation of the queries based on an ontology is a means for providing conceptual answers formed by ontology concepts. Evaluation of an ontology query can thus be considered as an intermediate step in the evaluation of queries for documents. In this case, the "ontology" can be one of the following:

- **The initial ontology**, a set of external resources, e.g. WordNet plus a top-ontology,
- **The generative ontology**, defined as an infinite ontology that can be created by the initial ontology and a set of relations, or
- **The instantiated ontology**, a finite ontology formed by a generative ontology restricted to the concepts found in the set of documents,

[3] The presentation of Knowledge Retrieval in Section 6.3 is a modified rendering of the presentation of "*Quering by descriptions expressions*" given in [Bulskov *et al.*, 2004].

where the latter appears to be the most interesting as it covers, on the one hand, the domain of the given retrieval system, and on the other, reflects the content of the document base. The first two may however be useful for ontology engineers, domain experts, etc. in connection with domain modeling.

Aggregation for multiple concept queries can be done in several ways with one simple option being fuzzy union. This topic is further discussed in [Andreasen *et al.*, 2002; Andreasen, 2001; Yager, 2000].

The retrieval framework presented here can be extended to allow queries posed directly as expressions in the concept language, thereby using the concept language as the query language. Thus in place of the phrase "*Some black cat*" or a list of words query "*black, cat*" to support a description query "*cat*[CHR:*black*]".

Let us consider ontology queries and assume an instantiated ontology interpretation. The answer to a single concept query is then a set of concepts appearing in the (descriptions of) documents that are most similar to the query concept, which is identical to an interpretation where the concept query is the expansion function (6.3), $CQ(c) = expansion(c)$. Consider, for example, the query $CQ =$ "*cat*[CHR:*black*]" evaluated using $expansion_{\alpha}(CQ)$, where α is a threshold limiting the set of similar concepts to those with a membership grade (i.e. similarity) $\geq \alpha$. The query is evaluated in the ontology shown in Figure 6.1, resulting in the following similar concepts:

$$\begin{aligned} similar_{0.6}(cat[\text{CHAR}{:}black]) = \; & .90/cat[\text{CHR}{:}black]+ \\ & .73/animal+ \\ & .67/dog+ \\ & .63/poodle \end{aligned}$$

This type of querying may be applicable in cases where the user has knowledge about the ontology and the database content and has a rather specific intention. Without knowledge about the ontology, however, it may be difficult in any case to pose such concept queries. Moreover, only brief knowledge about the database content would probably often give unsatisfactory or empty answers to posed queries.

Another argument that motivates concept queries is that posing a natural language query means letting go of control and the responsibility for a satisfactory conceptual representation to the system. With a concept query, the user gains control over the interpretation of the intention of the query, but may face problems with expressing queries for knowledge in the concept language. For instance, if the user is interested in knowledge about "*colored dogs in general or any specific type of colored dogs*", then the query is problematic because it cannot be expressed using pure concept language.

With special attention to experienced users and domain experts, it appears that there is a need for a query language with more expressiveness concerning "*navigating*" the ontology. For instance, a document about "*a very large black dog*" will only belong, to some (if any) extent, to the answer on the query "*large pet*" due to the similarity based evaluation of queries.

6.4 Query Language

A concept as a query maps to a set of similar concepts, while similarity is influenced by distance in the ontology. The extension to the concept language introduced here is specialization/generalization operators to cope with a quite useful notation for disjunctions along specialization and/or generalization in the ontology, thus avoiding reduced similarity over paths of specialization and/or generalization[4].

Given the concept language $\mathcal{L}$ based on the set of atomic concepts A and the set of semantic relations R, as described in Chapter 4, we define an extension of $\mathcal{L}$ to a query language $\mathcal{QL}$ as follows:

- $\mathcal{L} \subseteq \mathcal{QL}$
- $* \in \mathcal{QL}$
- if $c \in \mathcal{L}$ then $c_> \in \mathcal{QL}$ and $c_< \in \mathcal{QL}$
- if $c \in \mathcal{QL}$, $r_i \in R$ and $c_i \in \mathcal{QL}, i = 1, \ldots, n$
  then $c[r_1 : c_1, \ldots, r_n : c_n] \in \mathcal{QL}$

The interpretation of this extended language is the following. $*$ denotes any well-formed concept in $\mathcal{L}$. $c_>$ denotes any specialization of c, while $c_<$ denotes any generalization of c. A query involving the operators $<, >$ and $*$ can be considered a disjunctive query over the set of denoted concepts.

With the ontology in Figure 6.1, $dog_<$ denotes all of $\{$*dog*, *animal*, *anything*$\}$, while $dog_>$ denotes all of $\{$*dog*, *poodle*, *poodle*[CHR:*black*]$\}$. The set of denoted concepts for a query is obviously the crisp answer to the query when evaluated in the ontology. Thus, a query like "*Do we have dogs*", with the interpretation "*Give me a dog or some specialization of that*" can be expressed in the query $dog_>$ and the answer provides a conceptual description of the kinds of dogs that are currently contained in the database without specification of actual dogs and that are without cardinalities. The answer will read something to the effect of "*We have poodles in black color*".

Also with the ontology in Figure 6.1, *cat*[CHR:$black_<$] denotes all of:

$$\{cat[\text{CHR:}black], cat[\text{CHR:}color], cat[\text{CHR:}anything]\}$$

[4] The presentation of Query Language in Section 6.4 is a minor modified rendering of the original presentation given in [Bulskov *et al.*, 2004].

Concepts that are not part of the ontology such as $animal[\textsc{chr}{:}black]$ can, of course, also be used in queries with $animal[\textsc{chr}{:}black_{<}]$ which denotes:

$$\{animal[\textsc{chr}{:}black], animal[\textsc{chr}{:}color], animal[\textsc{chr}{:}anything]\}$$

One reasonable question related to the introduction of specialization/generalization-queries is the extent to which such aspects are already covered by the pure concept language. How is an expression such as $animal[\textsc{chr}{:}*]$ necessary for representing "*Animal characterized by just anything*"[5], when $animal[\textsc{chr}{:}\top]$, which basically denotes the same thing, can already be expressed. The most important argument for the extension is that we have to cope with the side-effects from introducing similarity, and also especially consider graduated decreases in similarity over longer paths of specialization.

All concept queries expressed with the query language $\mathcal{QL}$ can naturally be used in document retrieval as well, since it would just denote multi-concept queries. It would therefore also make sense to expand concept queries and then include similar concepts in the surroundings of the specialization/generalization paths defined by the concept queries. For instance, the expansion of the queries "*animal*" and "$animal_{>}$" with respect to the instantiated ontology in Figure 6.1 is:

$$\begin{array}{lll}
similar_{0.5}(animal) = & 1,00/animal+ & similar_{0.5}(animal_{>}) & 1.00/animal+ \\
 & 0.93/dog+ & & 1.00/dog+ \\
 & 0.93/cat+ & & 1.00/cat+ \\
 & 0.90/poodle+ & & 1.00/poodle+ \\
 & 0.87/cat[\textsc{chr:black:}]+ & & 1.00/cat[\textsc{chr:black:}]+ \\
 & 0.86/poodle[\textsc{chr:black:}]+ & & 1.00/poodle[\textsc{chr:black:}]+ \\
 & 0.60/anything+ & & 0.60/anything+ \\
 & 0.50/color & & 0.60/black+ \\
 & & & 0.50/color
\end{array}$$

where it is shown that all the concepts on the specialization path form a set of disjunctive concepts, thus giving 1.00 as the degree of similarity.

6.5 Surveying, Navigating, and Visualizing Domain Knowledge and Information Content

Instantiated ontologies can describe the domain of any given subset of concepts, i.e. restrict a general ontology to a set of instantiated concepts appearing, for instance, in a given set, in a document, or in a set of documents[6].

[5] Note that $animal[\textsc{chr}{:}*]$ and $animal[\textsc{chr}{:}\top_{>}]$ are equivalent.

[6] The presentation given in Section 6.5 is a minor adjusted rendering of the original presentation given in [Andreasen *et al.*, 2005a].

As such, the instantiated ontology can be applied within navigation, surveying, and the visualization of the topics covered by the domain in question. Obviously, ontology engineers can benefit from the possibility of visualizing the knowledge held by the ontology in use when maintaining the system. However, since the restriction can be from any set of concepts, the instantiated ontologies can visualize the semantics of queries, found documents, the relation between a query and a given document, etc., and thus, also be an improvement for normal users.

Consider the following example of a document collection with the following four instantiated concepts:

$$I = \{stockade[\text{CHR}{:}old], rampart[\text{CHR}{:}old], church[\text{CHR}{:}old], palisade\}$$

and a combination of WordNet, SUMO, and MILO as the global ontology.

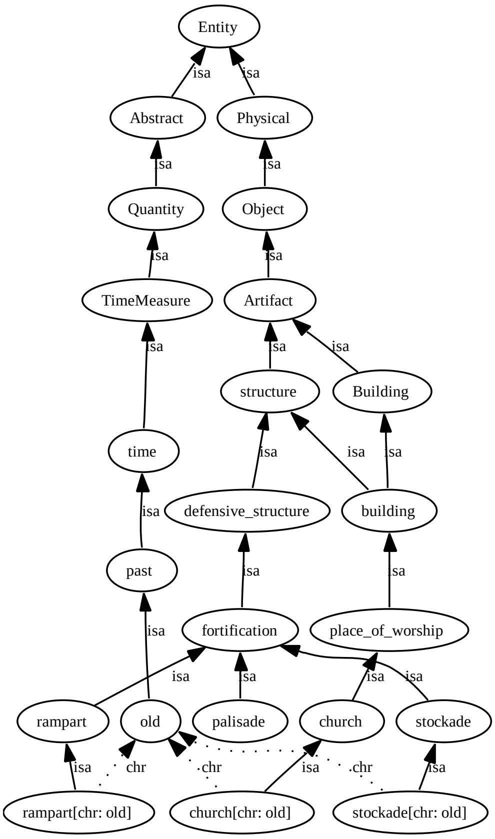


Figure 6.3: *A simple instantiated ontology, based on WordNet, SUMO, MILO, and the four concepts* $stockade[\text{CHR}{:}old], rampart[\text{CHR}{:}old], church[\text{CHR}{:}old], palisade$

The instantiated ontology reveals two different aspects covered by the document collection:

1. Different kinds of fortifications and
2. A place of worship.

On a more general level, the instantiated ontology describes buildings and the abstract notion of something dating back in time.

As one can see, there are concepts present in the instantiated ontology that are either very abstract or not part of an everyday vocabulary. When the user tries to form a general view of the document collection, such concepts could possibly lead to confusion and could/should therefore be removed from the visualization by utilizing the notion of *familiarity* as described in [Beckwith *et al.*, 1993]. Familiarity is defined using the correlation that exists between frequency of occurrence and polysemy. Using the *Collins Dictionary of the English Language*, an integer exists that is associated with every word form in the lexicon and that represents a count of the number of senses the word form has when it is used as a noun, verb, adjective, or adverb [Beckwith *et al.*, 1993].

This information can then be used to eliminate all concepts from the visualization of the instantiated ontology that have a familiarity lower than a certain threshold.

Another use of instantiated ontologies is for visualizing user queries. When users pose queries to the system using polysemous concepts, the instantiated ontology constructed from the query can be used to visualize the different senses known to the system. If, for example, a user poses a query $Q = \{bank, huge\}$, then the system cannot use the concept *huge* to resolve the context/disambuiguate *bank*, since *huge* can be used in connection with different senses of *bank*.

One possible way to incorporate the knowledge visualized is to let the user identify the intended sense from the visualization, and then use the disambiguated concepts in the query evaluation.

6.6 Summary and Discussion

In this chapter, we have introduced a number of different ideas related to the evaluation of queries in an ontological retrieval framework.

The first part of this chapter concerns *semantic expansion*, a way to introduce the similarity functions introduced in chapter 5 in the query evaluation. Two different expansion methods, object and query expansion, are discussed, while the latter was chosen as it offers maximal flexibility. The *semantic expansion* can be seen as a softening of the interpretation of concepts by adding relevant aspects to the query description. This will lead to increased recall and the assumption is that this in turn will lead to better answers. Obviously, one consequence is reduced precision, but this problem can be eliminated by using inherent modification mechanisms of the evaluation principle, for instance, the ranking of results.

The computing of the query expansion can be either static or dynamic. Static means that the computation of the similarity between concepts used in the system is done in advance, while dynamic means that the computation is done at query time. A simple static approach where the similarity between concepts is captured in a *similarity matrix* is recommended. Naturally, a static approach reduces flexibility as some of the constraints are bound, but by using so-called partial pre-processing, some of the flexibility can be regained. Dynamic expansion, on the other hand, offers maximal flexibility, but is more time consuming. A dynamic expansion similar to the spreading-activation algorithm used in semantic networks to determine least upper bounds between two or more concepts is introduced. The idea of this function is to confine the traversing of the ontology to the concepts we actually want to include in the expansion. Hence, only a small fragment of the concepts would serve as a softening of the interpretation of a given concept, while the rest would only contribute with noise in the query evaluation. Noticeably, finding the line of demarcation between the amplificational and the confusing concepts is a key issue. However, to establish whether a given expansion is supportive or deceptive for a given user is difficult, since similarity is rather subjective and most likely needs to be determined empirically for a given system or left open to the users by providing options to justify parameters of the expansion.

The second part of this chapter deals with the query evaluation framework in response to a user request. The major issue in information retrieval is to provide references to a set of documents that is likely to contain the information desired by the user. This process implies several sources of imprecision. First, the users may be looking for information they do not actually know they are looking for, and it is questionable whether the information needed can be reflected exactly in the query. Hence, the submitted query will be an imperfect expression of the information need. Second, descriptions of the content are extracted from documents in the indexing process to produce the systems representation of documents. Despite sophisticated indexing techniques, this representation will only comprise a partial and imprecise characterization. Finally, the retrieval process should establish a relationship between imprecise information needs and imprecise document descriptions to determine whether a document is relevant or not. A number of different approaches have been used to cope with the inexact representation of documents and queries (see Chapter 2 for an outline of some of these approaches).

The motivation for selecting the fuzzy set retrieval model is the intrinsic support of relevance as a multi-valued variable. This model grades documents to be more or less relevant and the uncertainty is thus an inherent part of the decision-making issue. Furthermore, it takes into account, in a natural way, the different contributions of domain concepts for the document and query characterizations, which is reflected in a grading of the relevance of documents to a given query.

A simple fuzzy query evaluation is introduced first, where a fuzzy set describing the query and fuzzy sets describing documents are used to determine the relevance of documents to a query. A best match Boolean query evaluation is a special case of the simple fuzzy query evaluation where the assignment of concepts to documents is binary, i.e. the description model is binary. However, even through the simple fuzzy query evaluation generalizes the Boolean model; it is not expressive enough to capture the intrinsic structure of the conceptual descriptions (Ontolog expressions). In order to include this intrinsic structure in the evaluation, we need to evaluate each element of queries separately and let the relevance be an aggregate over these results. For this purpose, order weighted averaging and hierarchical aggregation are introduced, where the latter uses the former to group queries into hierarchies, hence introducing nested query evaluation.

The final part of this chapter concerns different obvious prospects for introducing ontological knowledge in the retrieval process. First, the idea of *knowledge retrieval* is presented. The objective of information retrieval is retrieving documents, but the introduction of additional knowledge, external to the documents, gives good reason for querying the knowledge covered by the indexing of information in a retrieval system, e.g. the instantiated ontology. The introduction of ontologies makes this kind of querying even more interesting as it gives opportunities for questioning knowledge “hidden” in the mapping between the information in the documents and the information in the ontologies. Second, a query language, an extension to the concept language, is introduced to support specialization/generalization operators to cope with a quite useful notation for disjunctions along specialization and/or generalization in the ontology. Finally, surveying, navigating, and visualizing within the instantiated ontologies is briefly described. Instantiated ontologies describe the domain of a set of instantiated concepts, and thus can be applied within the navigation, surveying, and visualization of the topics covered by the domain.

Chapter 7

A Prototype

The purpose of this chapter is to sketch the structure and implementation of an experimental prototype system that incorporates ideas presented and discussed in previous chapters. The prototype, which is English language based, is inspired by a series of Danish language prototypes developed in the OntoQuery project and considered the first step in a more advanced implementation approach towards a final prototype for real life experiments. The prototype is intended to be the foundation for the evaluation and testing of the three main ideas introduced in this thesis, ontological indexing, ontological similarity, and fuzzy information retrieval.

The ontological basis of the prototype is a manipulated version of WordNet 2.1 (described in Section 3.5.3), where only nouns, adjectives and adverbs are considered. In addition, we use only a fragment of the relations present in WordNet in order to establish the simplest useful prototype ontology.

The document base used in the first version of the prototype encompasses sentences from the SemCor corpus [Miller *et al.*, 1994]. This corpus has the advantage that all words are tagged by part of speech, and many of the content words are lemmatized and sense-tagged according to WordNet. A document base therefore exists that can serve as test data with respect to part of speech tagging and word sense disambiguation, and which can be selected for use in place of the tagging and disambiguation modules in the prototype. The manually established SemCor knowledge has a high validity and is therefore well suited for isolated testing of the generation of descriptions.

In the prototype, we distinguish between a database and a knowledge base. The former includes documents and descriptions of these, while the latter contains knowledge, mainly in the form of an ontology and dictionaries.

The prototype includes two main components - description generation and query evaluation. The descriptions are generated when loading documents to the prototype database and when interpreting queries posed to the system. The generation of descriptions consists of a shallow natural language processing, an extraction of descriptions, and a mapping of extracted concepts into the prototype ontology. The query evaluation uses query expansion to incorporate ontological similarity and a hierarchical generalized fuzzy retrieval model to evaluate the comparison of query and document descriptions, i.e. reasoning within the description language Ontolog, in order to retrieve the documents matched by queries.

In the following, the overall design considerations for the prototype are touched upon first, followed by a discussion of the establishment of the prototype ontology and a short description of the document base. Subsequently, the generation of descriptions is examined, and finally, considerations about the evaluation process are discussed.

7.1 Design Considerations

In view of the fact that most of the ideas introduced have to be adjusted empirically, the prototype has to be flexible with regard to modifications. The method used in this prototype to include a high level of changeability means dividing the overall functionality into smaller pieces with well-defined interfaces and activities, which we call *modules*. For that purpose, the object-oriented design paradigm is used, since one important facet of effective object-oriented design is encapsulation. Encapsulation hides the inner details of the modules from the outside, and common interfaces define how to use the modules and access their information. For example, consider semantic expansion, a functionality that returns a number of similar concepts to a given concept. The interface to this functionality is very simple, while the underlying computation is rather complex. Any change in the underlying computation of this function should still accept the same type of input and produce the same type of output, even through the computation could be completely different. Hence, the change of the substance of a module does not influence the prototype in general.

A module can depend of other modules, thus forming a design structure of nested modules. Modules in the prototype communicate by use of their common interfaces and information is exchanged between modules by a set of streams. The streams are data structures that hold the information necessary for modules to communicate, and are designed by use of the object-oriented design paradigm. In contrast to the modules, changing the substance of the streams is normally not cost free, since the modules are designed to comply with the information held by the streams and their interfaces.

Figure 7.1 shows the overall architecture of the prototype system with its two main modules, *indexing* and *evaluation*, and their connections to the data and knowledge bases.

The *indexing module* handles the input to the prototype system, which is

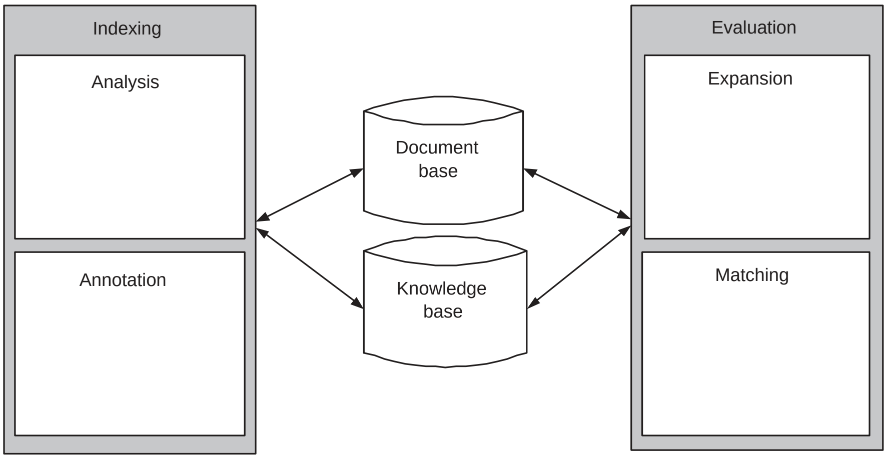


Figure 7.1: *A picture of the overall architecture of the prototype system*

either documents or queries. This module performs the analysis and annotation of the input, and is the union of an *analysis module*, which analyses the content of the input, and a *annotation module*, which extracts information from the analyzed input and creates the descriptions of the inputted objects. The reason for subdividing into two minor modules is due to considerations regarding the flexibility described above, as each of these modules can be changed without having to change the other one. The point of joining the modules of the description generation into a single module, the *indexing module*, with a common interface and well-defined behavior is to encapsulate all the inner details of the process.

Further treatment of the input in the system depends on whether it is documents or queries. Documents are stored in the system by the description produced by the *indexing module*, while descriptions of queries are used in the query evaluation by the *evaluation module*.

The *evaluation module* is a module that joins the *expansion module* , which expands queries by use of ontological similarity and the *matching module* , which performs the matching of queries and documents into a single module. The primary functionality of the *evaluation module* is to produce a list of the stored documents matching the description of a given a query.

7.2 The Prototype Ontology

In order to create the basis ontology for the prototype, we first have to choose which of the relations in WordNet to include as part of the general ontology of the prototype. Descriptions in the prototype are composed of simple noun phrases formed into Ontolog expressions, while the word classes used from

WordNet are thus nouns, adjectives, and adverbs. Hence, the relations from WordNet that are necessary to include in the prototype system are the ones concerning these three classes only.

The obvious relation for nouns is the concept inclusion relation (the hyponym relation and its opposite - hypernym), since this is the main relation in the ontology. However, WordNet has a related relation called *instance hyponym* (and its opposite *instance hypernym*), which is (primarily) used for the relation between proper nouns and their hypernym, e.g. “Johns Hopkins” is an *instance hyponym* of the synset {*university*}. The *instance hyponym* relation is also included in the prototype ontology, and both the *hyponym* and *instance hyponym* are treated as *hyponym* relations in the prototype ontology, even through the hyponym and the instance relations, in a strict ontological sense, should be separated[1]. In an information retrieval context, the connection between proper nouns and nouns, and thereby the possibility of computing similarity, is more important than the strict interpretation of the difference between these relations.

Another important relation concerning nouns in WordNet is the part-whole relation, called *meronomy*, e.g. “cell” is part of “organism” and “Japan” is part of “Asia”, and its opposite *holonymy*, e.g. “organism” has “cell” and “Asia” has “Japan” as a part. The part-whole relation forms a hierarchy, likewise the concept inclusion relation, and would therefore be useful as part of the similarity measure. However, the inclusion of the part-whole relation is postponed to a later version of the prototype, as the close relatedness to the concept inclusion relation (it is also a ordering relation) could introduce side effects, especially concerning the deduction of weights for the semantic relations. Furthermore, an overlap exists in WordNet between the part-whole relation and the concept inclusion relation, e.g. “head” and “arm” are both part of the concept “body” (part-whole) and subsumed by the concept “body part” (concept inclusion).

The modifiers, adjectives and adverbs are not part of the concept inclusion hierarchy of nouns. The semantic organization of adverbs in WordNet is simple and straightforward since they only have a derived from relation to the words they are derived from (in most cases adjectives). Hence, they do not form any structure of their own, but participate in the structure solely through the *derived from* relation.

Adjectives, on the other hand, are divided into two groups, descriptive and relational. The latter is derived from words in other word classes (primarily nouns). The relational adjectives are, like adverbs, not part of an independent structure, but participate in the structure through the *derived from* relation, while the ones derived from nouns are therefore connected to the concept

[1] See e.g. [Gangemi *et al.*, 2003] for a discussion of the problems related to the non-strict use of hyponym and instance relations in ontologies in general.

inclusion hierarchy.

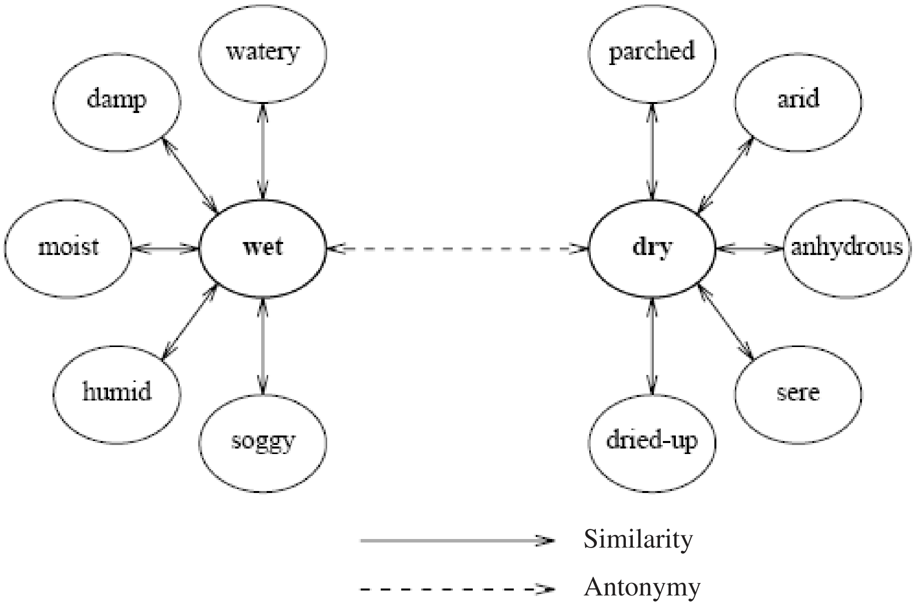


Figure 7.2: *Adjective antonym cluster with "wet" and "dry" as heads*

Descriptive adjectives are organized in so-called *antonym clusters*, as shown in Figure 3.14 in Section 3.5.3 and repeated here in Figure 7.2. The *heads* in an *antonym cluster* are the two synsets related by the *antonymy* relation in the figure on synsets, "wet" and "dry". These *heads* can have a number of similar *satellite* synsets connected through a *similar to* relation. In most cases *satellite* synsets are specializations of the head in the sense that the *satellite* synsets can only modify a subset of the nouns that can be modified by the *heads*.

These *antonym clusters* form small hierarchies of related adjectives that can be used when measuring similarity, e.g. a pair of compound concepts modified by adjectives from the same half-cluster[2] should be considered more closely related than if the same pair of concepts were modified by adjectives from different clusters.

Adjectives are used in the prototype via the inclusion of the *antonym clusters* (or more precisely the *antonym half-clusters*) and the *derived from* relation to nouns.

Consider the small instantiated ontology in Figure 7.3, composed from a general ontology based on the considerations described above, and restricted

[2] Only the *similar to* relation in the *antonym cluster* should contribute to similarity, not the *antonymy* relation, thus the notion of a half-cluster.

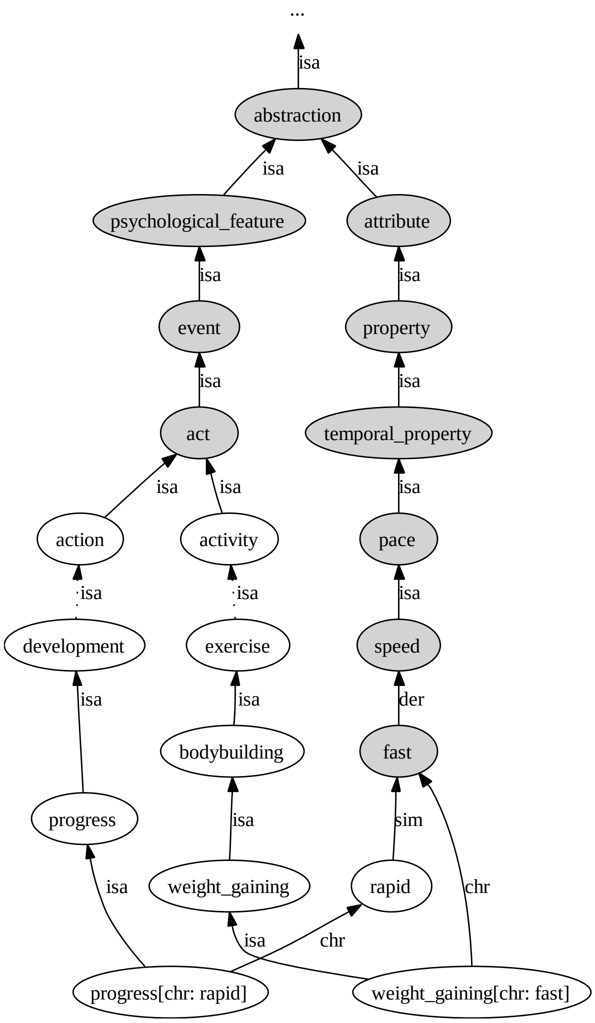


Figure 7.3: *The instantiated ontology restricted by the concepts "progress*[CHR:*rapid*]*" and "weight_gaining*[CHR:*fast*]*" with common nodes shaded gray. The dotted* ISA *arrows from "development" to "action" and from "exercise" and "activity" indicate that some concepts have been omitted in the visualization of the ontology*

by the concepts *progress*[CHR:*rapid*] and *weight_gaining*[CHR:*fast*]. Without the introduction of the *similar to* (SIM), these two compound concepts would only be compared by the concept inclusion hierarchy, as the concepts "rapid" and "fast" would not be connected. Due to the SIM relation, they are now connected; "rapid" is satellite to "fast", and due to the DIR relation they are also connected to the concept inclusion hierarchy and can thus contribute to the similarity measure as intended.

7.3 The Document Base

The document base in the prototype is the semantically tagged corpus SemCor [Miller *et al.*, 1994], created by Princeton University. The corpus is a subset of the *Brown Corpus of Standard American English* [Francis and Kucera, 1964], which contains almost 700,000 running words (20,000 sentences), and is composed of 352 texts. In 186 of the texts, all the open class words (nouns, verbs, adjectives, and adverbs) are annotated with part of speech, lemma, and sense, while in the remaining 166 texts only verbs are annotated with lemma and sense.

Figure 7.4 shows an example of the SGML format used to define sentences in SemCor. The <*s*> tag denotes a sentence and <*wf*> tags the word forms for the words in the sentence. The attributes *wnsn* and *lexsn* are the mapping of the words into WordNet, while the attribute *pos* is the part of speech and *lemma* is the lemmatization of the word.

```
<s snum=57>
<wf cmd=done pos=NN lemma=branch wnsn=1 lexsn=1:14:00::>Branch</wf>
<wf cmd=done pos=NN lemma=office wnsn=2 lexsn=1:14:01::>Offices</wf>
<wf cmd=done pos=VB lemma=be wnsn=3 lexsn=2:42:05::>are</wf>
<wf cmd=done pos=JJ lemma=located wnsn=1 lexsn=5:00:00:settled:01>located</wf>
<wf cmd=ignore pos=IN>in</wf>
<wf cmd=done pos=JJ lemma=other wnsn=1 lexsn=3:00:00::>other</wf>
<wf cmd=done pos=JJ lemma=large wnsn=1 lexsn=3:00:00::>large</wf>
<wf cmd=done pos=NN lemma=city wnsn=1 lexsn=1:15:00::>cities</wf>
<punc>.</punc>
</s>
```

Figure 7.4: *A sentence in the SemCor SGML format*

7.4 The Indexing Module

The main goal of the indexing module is to create descriptions of the documents and queries. The indexing module is the first part of the overall processing of documents and queries in the prototype. The description generation is handled by the *indexing module* and consists of a sequence of smaller processes. These processes are enclosed by the two internal modules, analysis and annotation. The aim of the analysis is to recognize noun phrases and to have the annotation perform the extraction of the descriptions. The processes required in order to create the descriptions are divided between these two modules in the following way:

- Analysis module

  - Tokenization
  - Part of speech tagging
  - Noun phrase recognition
- Annotation module
  - Concept extraction
  - Word sense disambiguation

and each of these processes is encapsulated in independent modules that can be changed without requiring rewriting the *indexing module*.

Tokenization is the simplest part of the indexing process and serves solely as a preprocessing to the part of speech tagging in order to prepare the input for the tagging. Naturally, any conversion specific to a particular part of speech tagger should be performed by that process itself, but since the input can be both documents and queries, some minor transformations may be required. Any conversion required of the input to simple text can also be performed by the *tokenization* process, e.g. conversion of HTML formatted input, but most likely such kinds of conversions would normally require handling before the input is given to the indexing module in order not to limit the indexing module to a specific kind of document.

Part-of-speech tagging is the process of assigning part of speech tags to words. Part-of-speech tagging is harder than just having a list of words and their parts of speech, because some words can represent more than one part of speech at different times, e.g. the word "file" can act as a substantive or a verb. *Part-of-speech tagging* was earlier considered an inseparable part of natural language processing, because there are certain cases where the correct part of speech cannot be decided without understanding the semantics or even the pragmatics of the context. However, in the mid 1980s when researchers started using statistical techniques for natural language parsing hidden Markov models were used successfully to disambiguate parts of speech. Hidden Markov models involve counting cases and making a table of the probabilities of certain sequences. For example, when an article appears, the next word is a noun 60% of the time, an adjective 30%, and a number 10%. With this type of knowledge, a program can decide, for instance, that "can" in "the can" is far more likely to be a noun than a verb or a modal. It is worth remembering that merely assigning the most common tag to each known word and the tag "proper noun" to all unknowns means approaching 90% accuracy because many words are unambiguous [Charniak *et al.*, 1996]. Nowadays, there is a wide range of different statistical techniques for *part-of-speech tagging*, where some of the current major algorithms include the Viterbi algorithm, transformation-based tagging (Brill), the Baum-Welch algorithm (also known as the forward-backward algorithm), and hidden Markov models [Manning and Schütze, 1999].

The method used in the prototype is the transformation-based part-of-speech tagger by Brill [1997]. However, since the quality of the different statistical techniques for part-of-speech tagging is very similar, any of the above-mentioned techniques could have been used. Obviously, an alternative solution is to use *boosting*, the idea of combining several (moderately accurate) methods into a single, highly accurate approach, hence combining several of the above techniques.

In the prototype *noun phrase recognition* is a process where noun phrases are extracted from part-of-speech tagged text. Noun phrases fluctuate from single words to complete sentences and the computational complexity of recognizing noun phrases thus varies depending on the kind of noun phrases that need to be recognized.

Noun phrase recognition is controlled by a grammar that defines the structure of the recognizable phrases. Figure 7.5 shows the noun phrase grammar used in the prototype. This grammar limits the phrases to the nominal plus pre-modifiers, i.e. noun-phrase chunks extending from the beginning of the constituent to its head, plus simple prepositional phrases, where noun phrases are combined with prepositions.

```
nounPhrase           ::=  preDeterminer?
                          determiner?
                          cardinalNumber?
                          adjectivePhrase?
                          nominal+
                          prepositionalPhrase;
prepositionalPhrase  ::=  preposition nounPhrase;
preposition          ::=  IN;
preDeterminer        ::=  PDT;
deteminer            ::=  deteminerTag;
deteminerTag         ::=  DT;
cardinalNumber       ::=  CD;
adjectivePhrase      ::=  adverb+ adjective* |
                          adjective* |
                          (adjective ",")* adjective ","?  CC adjective;
adverb               ::=  RB | RBR | RBS;
ajective             ::=  JJ | JJR | JJS;
nominal              ::=  NN | NNP | NNS | NNPS;
```

Figure 7.5: *The Noun Phrase Grammar in EBNF*

For example, the following text fragment, *The black book on the small table.* would be assigned the following tags by the Brill tagger:

The/DT *black*/JJ *book*/NN *on*/IN *the*/DT *small*/JJ *table*/NN and consist of the two simple noun phrases "black book" and "small table" combined by the preposition "on". Thus, the output from the noun phrase recognition given the grammar in Figure 7.5 is:

⟨PP [NP *The*/DT *black*/JJ *book*/NN] *on*/IN [NP *the*/DT *small*/JJ *table*/NN]⟩

where prepositional phrases are marked by angle brackets and noun phrases by square brackets.

The final part of the indexing module concerns the generation of descriptions and word sense disambiguation, where the former extracts concepts (noun phrases) and the latter maps them into the ontology.

The recognition of the relations that should be assigned to different prepositions is very complicated due to a high degree of ambiguity[3]. The generation of compound concepts from prepositional phrases in the prototype is restricted to the preposition "on", as this is one of the least complicated prepositions. The "on" preposition is by default assigned the location relation (LOC), e.g. in the above example, the concept generated is:

$$book[\textsc{chr}{:}black, \textsc{loc}{:}table[\textsc{chr}{:}small]]$$

where the default assignment is the right interpretation. However, the preposition "on" is also used in natural language to denote position in time, e.g. *the meeting on Monday*, while the semantic relation to be used in these types of cases is the temporal relation (TMP). The solution to this problem is to assign the temporal relation whenever the constituents can be determined as temporal concepts by use of the ontology.

Word sense disambiguation is done by use of the information in WordNet about sense frequencies, i.e. the most frequent sense used for a given word. Naturally, for the sentences in SemCor, the mapping from the corpus can also be used.

After the descriptions are generated, they are used for querying (queries), or they are added to the index in the database and to the instantiated ontology in the knowledge base (documents).

7.5 The Evaluation Module

The two main modules in the *evaluation module* are *expansion* and *matching*, where the former uses the weighted shared modes similarity measure to expand queries, and the latter performs the actual comparison between descriptions of queries and descriptions of documents.

[3] See e.g. [Quirk *et al.*, 1985b] for a description of interpretations of prepositions in the English language.

The technique used in the expansion process is similar to the semantic network algorithm *spreading-activation* described in Section 6.1.1. In the prototype, two thresholds are used to control spreading-activation, the degree of similarity and the cardinality of the set of expanded concepts, hence either the spreading stops, as no more concepts can fulfill the degree of similarity, or the number of expanded concept reaches the cardinality threshold. This expansion algorithm is efficient and adds no significant time overhead to the query evaluation, even through it is executed at query time on an instantiated ontology with 100,000 concepts. Obviously, the two thresholds could be adjusted in such a manner that the expansion would be very inefficient, e.g. the worst-case scenario is an activation of the complete ontology for each concept in the query.

In contrast to the above thresholds, are the thresholds used in the similarity function, which are not as straightforward. Each of the relations used in the ontology has a weight assigned denoting the cost of using that relation as part of the path measuring the distance between concepts in the ontology. Furthermore, the similarity function:

$$sim(x,y) = \rho\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(x)|} + (1-\rho)\frac{|\alpha(x) \cap \alpha(y)|}{|\alpha(y)|} \tag{7.1}$$

defined in Section 5, has the parameter ρ, which determines the degree of influence from generalizations and specializations. The similarity for specializations → 1 when $\rho \rightarrow 1$, and the similarity for generalizations → 1 when $\rho \rightarrow 0$. The similarity function is symmetric when $\rho = 0.5$, which is undesirable due to the arguments given in Chapter 5.

		entity	artifact	vehicle	car	jeep
$\rho = 1.0$	vehicle	0,33	0,67	1,00	1,00	1,00

		entity	artifact	vehicle	car	jeep
$\rho = 0.0$	vehicle	1,00	1,00	1,00	0,75	0,60

		entity	artifact	vehicle	car	jeep
$\rho = 0.8$	vehicle	0,47	0,73	1,00	0,95	0,92

Figure 7.6: *Examples of how different values for ρ influence the similarity measure for the concept "vehicle" in a path from the concept "jeep" upwards to the top-most concept "entity"*

Figure 7.6 shows how ρ influences the similarity measure for the concept "vehicle" in a path from the concept "jeep" upwards to the top-most concept "entity". The two first examples show the extremes when $\rho = 1$ and $\rho = 0$. The last example shows the chosen default value for ρ, which substantiates the intuition that a specific answer to a general query is better than a general answer to a specific query. Furthermore, distance matters both for generalizations and specialization, e.g. sim("vehicle", "car") > sim("vehicle", "jeep") and sim("vehicle", "artifact") > sim("vehicle", "entity").

The weights for relations and ρ must be resolved empirically, for instance, by use of human similarity judgments as described in Section 5.5, where the weights for the relations "characterized by" (CHR), "caused by" (CBY), and "with respect to" (WRT) are determined to be 0.57, 0.30, and 0.36, respectively[4]. In this context, retrieval evaluation is another obvious manner to resolve the values for the weights and ρ, thus adjusting the values towards the best recall and precision. The major challenge in using the latter technique is that it is very difficult to isolate the consequences of changing the weight of a single relation, since the retrieval most likely is based on a set of different relations. One solution to this challenge is to use the technique of training using the backpropagation of errors, similar to how neural networks are trained [Masters, 1993], and to learn the weights and ρ through the retrieval evaluation by using a set of training data.

The indexing process contributes with more detailed descriptions of documents and queries; the benefit achieved from this is a more refined view of the information, and thereby the possibility of matching concepts, i.e. noun phrases, instead of words. The ontological knowledge is utilized in the prototype through the semantic expansion of queries.

Consider the query "a possible alternative", i.e. a search for information about possible alternatives in some context. This query is transformed into the concept *alternative*[CHR:*possible*] by the indexing process. With $\rho = 0.8$, $weight(\text{ISA}) = 1$, $weight(\text{CBY}) = 0.57$, and 0.95 as the weight for the adjective relations SIM and DIR. The semantic expansion of this concept is:

$$expansion(alternative[\text{CHR}{:}possible]) = \begin{array}{l} 1.00/alternative[\text{CHR}{:}possible]+ \\ 0.86/alternative[\text{CHR}{:}workable]+ \\ 0.83/alternative[\text{CHR}{:}real]+ \\ 0.78/alternative+ \\ \vdots \end{array}$$

where both *alternative*[CHR:*workable*] and *alternative*[CHR:*real*] are considered more similar than the generalization *alternative*. This is a consequence of the fact that attribution contributes to the similarity measure, and that the

[4] In the human similarity judgments comparison performed in Section 5.5, ρ is fixed to the value 0.5 in order to make the similarity measure symmetric, and thus comparable to the other similarity measures.

attributes, *workable*, *real*, and *possible* are similar. Figure 7.7 shows the instantiated ontology restricted by the concepts *alternative*[CHR:*possible*] and *alternative*[CHR:*workable*], with their shared nodes shaded gray This illustration also adequately underpins the benefit of including the two adjective relations, SIM and DIR, in the ontology, as the connection between *workable* and *possible* is due to the SIM relation and the connection of *possible* to the concept inclusion hierarchy is due to the DIR relation.

The last part of the evaluation process is the matching of document descriptions to the (expanded) description of queries. In the prototype, the fuzzy information retrieval model with hierarchical order weighted averaging aggregation is used. As mentioned in Section 6.2.1, one challenge raised by the introduction of fuzzy information retrieval is that, in general, when all concepts are assigned to every document, all documents have to be evaluated. This problem has to be handled in order to scale the retrieval model to fit huge document collections.

The simple optimization is to restrict the aggregation to the set of documents which possibly can satisfy a query, i.e. all the documents that match a least one of the concepts in the query. Strictly speaking, this problem is not solely bound to fuzzy information retrieval, because any model that supports "best match" evaluation has to deal with this problem. Normally, further optimizations can be handled by traditional optimization algorithms, e.g. database indexing, cost-based looping due to cardinalities, etc. However, the order weighted averaging aggregation requires an ordering of the matching weights for each document, which is not immediately consistent with these traditional algorithms.

Given a query $q = \{A, B, C\}$ with three concepts, the matching process uses the simple optimization first and narrows the set of documents to the subset P_Q with the documents that match at least one of the concepts in the query. A tuple $\{w_A, w_B, w_C\}$ is created for each document with a weight for each concept in the query denoting the compliance of the concepts with the document. The tuple is then ordered and the weighted averaging, and (possibly) importance weighting, are performed. The upper and lower bounds of the ordered weighted averaging *max* and *min*[5], and the simple mean comply partly with the traditional algorithms, due to the fact that *max* and *min* correspond to OR and AND, and as the simple mean does not require ordering.

Two specialized optimizations are introduced in the prototype to provide scalability to huge document collections, frequency-based matching and upper-bound evaluation. The frequency-based matching optimization divides concepts into two groups, normal and high frequency concepts, where the frequency refers to the number of documents to which the concept is assigned.

[5] Assuming that the T norms and T co-norms used for *min* and *max* correspond to the standard versions (see Section 2.1.5).

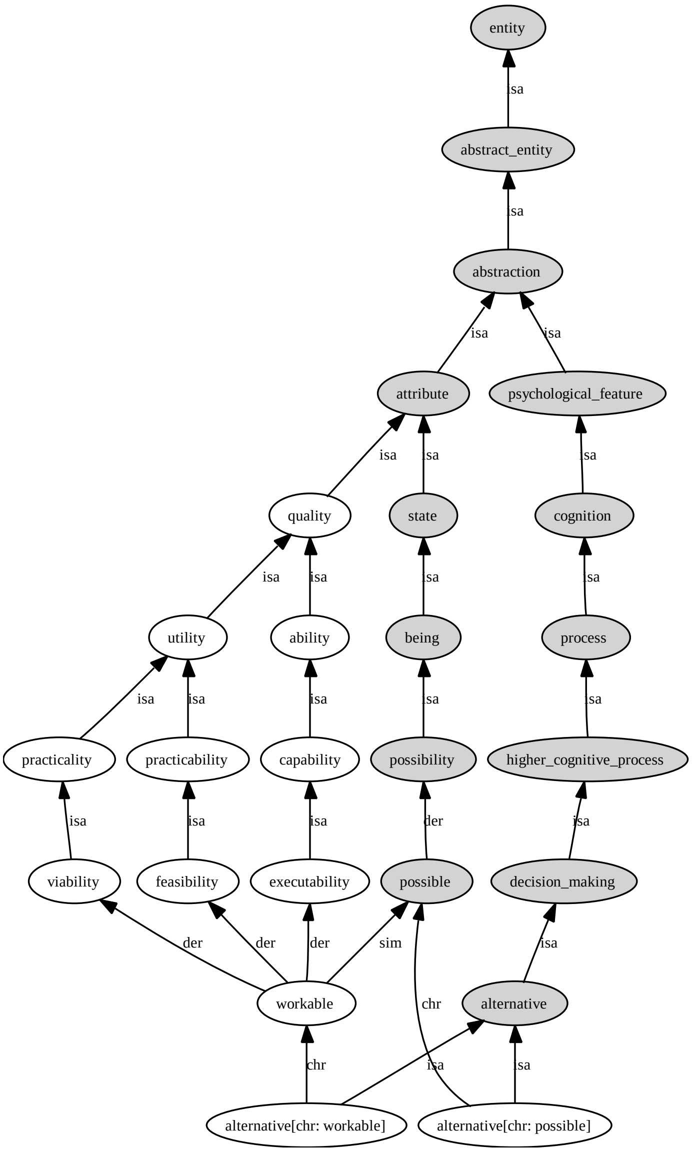


Figure 7.7: *The instantiated ontology restricted by the concepts alternative*[CHR*:possible*] *and alternative*[CHR*:workable*]*, with their shared nodes shaded gray.*

This partition is controlled by a threshold defining the line of demarcation between normal and high frequency concepts. Queries can then be put into the three following categories:

1. Queries composed of normal concepts,
2. Queries composed of high frequency concepts, and
3. Queries composed of both normal and high frequency concepts.

The first category is evaluated normally, while the second and third categories are handled differently. The second category consist solely of high frequency concepts and the possibility of finding documents that match all the concepts in the query therefore increases, hence the need of a "best match" evaluation is less important. Instead of starting with a "best match" evaluation, a strict AND match is performed, hopefully resulting in a reasonable set of documents. Otherwise, either the "best match" evaluation is performed automatically, or the user is advised and can make the decision whether to continue with the "best match" evaluation or rewrite the query. In the third category, normal and high frequency concepts are mixed. The evaluation is separated such that high frequency concepts are evaluated by a strict evaluation and the result of this evaluation is used to envelop the evaluation of the set of normal concepts.

The upperbound evaluation optimization is used for queries with more than two concepts and searches for the r top-ranking documents, where r is defined by the user. Each concept c_i in query q has an associated list of documents; the set of documents to which the concept c_i is assigned. The cardinality of the lists of documents is used as the first part of the optimization, since the lists are accessed sequentially from the smallest to the largest. The evaluation computes the retrieval status value for $j+1$ document lists, where j is the number of the previously processed document lists out of i. This differs from the normal evaluation, which uses a tuple with the weights from all the concepts in the query for the computation of the retrieval status value. The r top-ranking documents are now identified considering only the $j+1$ concept in the query. Without accessing the rest of the document lists, it is possible to exclude all the documents in this partial result that, given the best case scenario for all the succeeding lists, cannot be in the r top-ranking. If the excluded documents are found when computing the succeeding list, they can be skipped. Furthermore, the algorithm can be terminated if it is not the case that any document in the succeeding lists are among the r top-ranking documents, otherwise j is incremented and the algorithm continues.

Consider a query with three concepts and an ordered weighted averaging with the weighting vector $w = \{0.2, 0.3, 0.5\}$ (an evaluation between mean and

AND) in a retrieval model where concepts with binary values are assigned to documents. The retrieval status value of documents can then have only four different values:

$$\begin{aligned}
1 \times 0.2 + 1 \times 0.3 + 1 \times 0.5 &= 1.0 \\
1 \times 0.2 + 1 \times 0.3 + 0 \times 0.5 &= 0.5 \\
1 \times 0.2 + 0 \times 0.3 + 0 \times 0.5 &= 0.2 \\
0 \times 0.2 + 0 \times 0.3 + 0 \times 0.5 &= 0.0
\end{aligned}$$

denoting that the document matches three, two, one, or none of the concepts in the query. The last one is excluded by simple optimization, so only the first three are possible values. Given that the retrieval status value of d_r, the r'th ranked document, after accessing the first two document lists is 0.5; hence, the r top-ranked documents have a retrieval status value of ≥ 0.5. We then know that all the documents that received the value 0.2 cannot obtain a total value that is greater that the value of d_r after accessing the last document list, and thus cannot change the r top-most ranking. Furthermore, only the documents with the value 0.5 as a result of processing the two first lists are candidates for a r top-most ranking, since any documents not seen already, i.e. new documents in the last list, can only obtain a value of 0.2. Thus, only the documents with the value 0.5 from the processing of the two first lists that also appear in the last list must be gone through.

The hierarchical aggregation is put into action whenever the query is expanded, since we want to evaluate a query concept and its expansions separately. As a result, the aggregation over the expansion can differ from the overall aggregation, i.e. different linguistic quantifiers can be assigned to the different levels in the evaluation. Obviously, the optimizations described above can likewise be performed independently on each level of the evaluation.

Chapter 8

Conclusion

The aim of this thesis, as indicated in the beginning, was to investigate and discuss introduction of ontologies to information retrieval, and thereby the promotion of semantics in the document retrieval process. The main research question was how to recognize concepts in information objects and queries, represent these in the information retrieval system, and use the knowledge about relations between concepts captured by ontologies in the querying process. In the thesis we have focused on the following three main aspects related to the research question

1. recognition and representation of information in documents and queries and the mapping of this knowledge into the ontologies,
2. improvement of the retrieval process by use of similarity measures derived from knowledge about relations between concepts in ontologies, and
3. how to weld the ideas of such ontological indexing and ontological similarity into a realistic scalable information retrieval scenario,

and the conclusion first discusses and concludes on these three elements separately, after which an overall conclusion is given and perspectives that can be used as the subject of further work is indicated.

8.1 Ontological Indexing

The chosen **representation of descriptions** is a collection of expressions formed by the lattice-algebraic language Ontolog. Compound concepts are defined as the attribution of concepts by feature attachment, both as multiple and nested attributions. Examples of such compound concepts are *book*[CHR:*cheap*, CHR:*short*], and *book*[LOC:*table*[LOC:*school*]], respectively. ONTOLOG expressions can be mapped directly into the ontologies and serve as the foundation of the ontological indexing. In order to establish ontology-based indexing, the semantics from the text, documents and queries must be revealed, thus extraction applying natural language processing techniques has to be an inherent part of the indexing.

Two different views on **semantic analysis** are discussed. First, conventional approaches to the disambiguation of word classes and senses, and second, a model where the ambiguities are weighted proportionally to the probability of occurrence, hence ambiguity, are are part of the analysis. An obvious advantage of disambiguation is a reduction in the computational complexity. However, wrong choices in the disambiguation process may imply a serious negative impact on the result, since errors would cascade and, especially when expansion is involved, be self-perpetuating during the processing of the query.

In connection with the extraction of compound concepts, a decision has to be made as to which level it would be preferable to extract compound concepts, also considering which side effects the **compoundness** introduces in the query evaluation. Assume, for example, that indexing on the sentence level can parse and generate one unique compound concept per sentence, namely an expression that describes the full meaning of the sentence. From a linguistic point of view, this would probably be an optimal solution, but for the information retrieval process, it introduces some problems. A high level of compoundness implies the need for a more complex reasoning to resolve similarity and conflicts, when comparing descriptions, with the key issue in retrieval. For this purpose, descriptions in the form of a collection of aspects of content opens far more possibilities for matching than single compound expressions intended to capture full meaning.

Ontological indexing can be considered as a mapping from documents to concepts in the ontology. The set of all concepts found in a document collection can serve as a way to restrict a given general ontology to an **instantiated ontology** describing the domain defined by the set of the "instantiated" (recognized) concepts. Other applications for instantiations of general ontologies are also obvious. For instance, restrictions on the concepts in a given query where the user is asked to choose between different senses in an instantiated ontology can serve as a solution to disambiguation. Furthermore, instantiated ontologies can be used in the visualization of the similarity between documents. Restrictions on the concepts the documents are described by, visualizes their conceptual overlap.

8.2 Ontological Similarity

Similarity measures derived from knowledge about relations between concepts in ontologies are discussed based on a set of intuitive properties of similarity in relation to information retrieval, accompanied by a discussion of the methodologies used. Three different kinds of similarity properties are used in the discussion, basic, retrieval-specific, and structure-specific properties. The basic properties define a very abstract notion of similarity and are obeyed by all the measures. The most interesting property, besides basic properties, is the retrieval-specific property called generalization, which relates to the intention of queries – a specific answer to a general query is better than a general answer to a specific query. This indicates that in the ontology-based information retrieval context, the similarity measure cannot be symmetrical, and should somehow capture that the "cost" of similarity in the direction of the inclusion (generalization) should be significantly higher than similarity in the opposite direction of the inclusion (specialization). Most of the measures presented fail to comply with this property, as they are symmetric. A consequence of the use of symmetric measures in information retrieval is that the evaluation of queries based on these measures would accept documents equally concerning specializations and generalizations to the posed concepts.

The proposed **shared nodes** measure has the ability to include not only paths between concepts through the ordering relation, but all possible relations connecting a pair of concepts. This is, in our opinion, a significant step in the right direction and the need for this generalization of similarity measures can be confirmed by human similarity judgments. When asked about the similarity between, for instance, a "forest" and a "graveyard" compared to the similarity of the same pair of concepts modified by the attribute "scary", the similarity is judged to be significantly higher when they share an attribute commonly used in connection with both concepts. The major problem with the shared node measure, in relation to this, is that not all relations equally define concepts, hence the fact that though two concepts share an attribute, does not add to their definitions in the same manner as concept inclusion does. To remedy this, a generalized shared node measure, the **weighted shared nodes** similarity measure, is introduced.

As a general observation, we would furthermore like to emphasize that the modeling of similarity functions is far from objective. It is not possible to define optimal functions in either a general or a domain specific case. We can only endeavor to have flexible and parameterized functions based on obvious "intrinsic" properties with intuitive interpretations, and then adjust and evaluate these functions on an empirical basis. The tailoring of the proposed similarity function can be done according to specific needs regarding the structure and relations of the ontology covering the domain, or it can be done to accommodate specialized user needs.

The major challenge in utilizing a concept similarity measure in a query evaluation is to reflect the concept-to-concept similarity in a query-to-document similarity. Basically, this issue involves comparing descriptions in a way that respects the concept similarity, while the approach is to introduce an expansion applied to queries, as well as an aggregation principle to combine with description similarity.

8.3 Query evaluation

The **semantic expansion** of concepts into a set of similar concepts based on a measure that only reflects the similarity between pairs of concepts is computationally demanding, as it has to compare, in principle, the concept being expanded to all concepts in the ontology in order to select the set of most similar concepts. To remedy this, an expansion function based on the notion of spreading activation is introduced. The similarity measure is influenced by distance in the ontology and the closest related concepts are therefore the concepts in the "ontological surroundings" of the concept we want to expand. A spreading activation expansion does this by only traversing the "closest" concepts to obtain the set of concepts we want to include in the expansion.

A **generalized fuzzy set retrieval model** is used as the foundation of the query evaluation; while not commonly used in present information retrieval systems, it fits perfectly into the proposed ontology-based retrieval model. The primary motivation for selecting this model is the intrinsic support of relevance as a multi-valued variable. Hence, this model takes into account, in a natural way, the different contributions of domain concepts in the document and query characterizations, and reflects this in a grading of the relevance of documents to a given query.

The **ordered weighted averaging** aggregation principle used in the proposed retrieval model is very flexible, and especially the ability of modeling the aggregation by "linguistic quantifiers" is convenient when the model is extended into a hierarchical model.

Basically, **hierarchical aggregation** extends ordered weighted averaging to capture nested expressions. Thus, queries may be viewed as hierarchies, and the hierarchical aggregation is perfectly suited for this purpose. Other models could equally be set op in a hierarchical framework like this, e.g. the vector model, but the modeling of the aggregation by linguistic quantifiers makes the evaluation process much easier to understand than, e.g. different kinds of ranking functions based on correlation in a vector space.

While the objective of document retrieval systems is obviously documents, the introduction of additional knowledge, external to the documents, gives rise to a need for additional knowledge retrieval. The introduction of ontologies makes this kind of querying even more interesting as it gives opportunities for exploiting knowledge “hidden” in the mapping between the information in the documents and the information in the ontologies, i.e. querying instantiated ontologies. The **querying of the knowledge base** in the ontology-based retrieval system substantiates an alternative query language. For this purpose, a query language is introduced to support specialization/generalization operators to cope with quite a useful notation for disjunctions along the specialization and/or generalization in the ontology. Since knowledge retrieval can be seen as the first step in document retrieval, the extended query language can be used for document retrieval as well.

8.4 Summery

A number of the presented techniques are joined into a **prototype system** intended to be the foundation for the evaluation and testing of the three main ideas introduced in this thesis, ontological indexing, ontological similarity, and fuzzy information retrieval. The creation of this prototype underpins that the methodologies presented can be used to form an ontology-based information retrieval system, while empirical studies are needed in order to state whether these methodologies, also in the “real world”, can contribute with the improvements indicated throughout the thesis. Nevertheless, the theoretical foundation has been presented and we have shown that these theories can be transformed into practice.

8.5 Further Work

Since this thesis covers most of the internal elements of a complete ontology-based information retrieval system, it has various issues that are candidates for further work. The most obvious aspect is naturally empirical studies.

One interesting and challenging step in this direction would be to establish large-scale experiments to evaluate the overall idea of content-based search. The goal is therefore to promote, for example, the Text Retrieval Conference (TREC), which has become a common and much referred platform for the testing of large-scale information retrieval approaches. However, a number of minor experiments are needed before this can happen in order to validate and fine-tune the prototype. As described in this thesis, only some of the methodologies looked at are included in the prototype presented. The first step towards achieving the above goal would therefore be to establish a number of minor experiments in order to find ways to include and evaluate the remaining methodologies and select those best suited for achieving the overall refinements.

The method presented to achieve the ontological indexing is rather ad hoc and is primarily constructed on well-known techniques. The syntactic analysis, i.e. the extraction of simple noun phrases, is reasonably solid and is used in a number of different approaches, e.g. information extraction. Word sense disambiguation is also a well-established research area, but concerns an incredibly more complex problem, and thus the available techniques are less reliable. We have briefly mentioned a word sense disambiguation technique that uses semantic similarity to determine the senses. This technique could serve for testing similarity measures, for instance, by using the semantically tagged SemCor corpus as the document base in the prototype, since part of this corpus is mapped into WordNet. Another more untested element of ontological indexing is the identification of semantic relations. Some preliminary experiments have been done on the use of ontological information in connection with supervised machine learning approaches in order to identify semantic relations between noun phrases in sentences. This work has shown promising results and may turn out to be very important in the generation of descriptions in ontology-based information retrieval.

The proposed similarity measure called weighted shared nodes was briefly tested in this thesis by comparing it to human similarity judgments. The test indicated that human similarity testing can be fruitful, especially with respect to determining the weights related to the semantic relations in use. Further work should therefore include a more trustworthy experiment that can be used to determine the (initial) weights associated with the semantic relations. Another way to achieve these values is obviously to use retrieval evaluation, since obtaining the best possible quality in the retrieval process is the primary goal. This would of course require some kind of training data, i.e. a document base and a set of queries complete with the preferred results. TREC provides such data, which furthermore supports promoting the TREC conference. Given some test data, resolving the weights for the relations as well as the parameters that control similarity functions is very complicated. Inspired by the training of neural networks, one solution could be to use the backpropagation of errors to adjust to the best possible set of values for a particular set of training data, but obviously, most of the training algorithms used in machine learning could just as well be the foundation for such an approach.

One of the characteristics of the presented similarity measures relates to the use of corpus statistics. The *information content* approach by Resnik [1995] is one such measure that uses sense frequencies to attach weights to the senses in ontologies. A natural further development of the *weighted shared nodes* measure would be to attach similar knowledge to the shared nodes. That is, combine the structure of the ontology and the corpus statistics as a basis for deriving what concepts have in common.

Finally, an interesting perspective related to the generalized fuzzy retrieval model concerns modeling by “linguistic quantifiers”. Words used in queries that indicate quantifiers could contribute to linguistic quantifiers in aggregation expressions rather than to the compound concept expressions. Hence, users would then be able to modify the evaluation of queries, not by some fancy interface, but by the use of natural language.

Bibliography

[Abney, 1996] Steve Abney. Partial parsing via finite-state cascades. In *Workshop on Robust Parsing, 8th European Summer School in Logic, Language and Information*, pages 8–15, Prague, Czech Republic, 1996. 71

[Agirre and de Lacalle, 2003] Eneko Agirre and Oier Lopez de Lacalle. Clustering wordnet word senses. In Nicolas Nicolov, Kalina Bontcheva, Galia Angelova, and Ruslan Mitkov, editors, *RANLP*, volume 260 of *Current Issues in Linguistic Theory (CILT)*, pages 121–130. John Benjamins, Amsterdam/Philadelphia, 2003. ISBN 1-58811-618-2. 77

[Agirre and Rigau, 1996] Eneko Agirre and German Rigau. Word sense disambiguation using conceptual density. In *Proceedings of the 16th conference on Computational linguistics*, pages 16–22, Morristown, NJ, USA, 1996. Association for Computational Linguistics. 76

[Allen, 1984] James F. Allen. Towards a general theory of action and time. *Artif. Intell.*, 23(2):123–154, 1984. ISSN 0004-3702. 51

[Alonge *et al.*, 2000] Antonietta Alonge, Francesca Bertagna, Nicoletta Calzolari, Adriana Roventini, and Antonio Zampolli. Encoding information on adjectives in a lexical-semantic net for computational applications. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 42–49, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 78

[Andreasen and Nilsson, 2004] Troels Andreasen and Jørgen Fischer Nilsson. Grammatical specification of domain ontologies. *Data Knowl. Eng.*, 48(2):221–230, 2004. ISSN 0169-023X. 48

[Andreasen *et al.*, 2000] Troels Andreasen, Jørgen Fischer Nilsson, and Hanne Erdman Thomsen. Ontology-based querying. In *FQAS*, pages 15–26, 2000. 5, 70

[Andreasen *et al.*, 2002] Troels Andreasen, Jørgen Fischer Nilsson, Per Anker Jensen, and Hanne Erdman Thomsen. Ontoquery: Ontology-based querying of texts. In *AAAI 2002 Spring Symposium*, Stanford, California, 2002. AAAI. 5, 70, 135

[Andreasen *et al.*, 2003a] Troels Andreasen, Henrik Bulskov, and Rasmus Knappe. From ontology over similarity to query evaluation. In R. Bernardi and M. Moortgat, editors, *2nd CoLogNET-ElsNET Symposium - questions and answers: Theoretical and applied perspectives*, pages 39–50, Amsterdam, 2003. CoLogNET-ElsNET Symposium, Amsterdam: Elsevier Science. 90

[Andreasen *et al.*, 2003b] Troels Andreasen, Henrik Bulskov, and Rasmus Knappe. Similarity from conceptual relations. In Ellen Walker, editor, *22nd International Conference of the North American Fuzzy Information Processing Society*, pages 179–184, Chicago, Illinois, USA, 2003. NAFIPS 2003. International Conference of the North American Fuzzy Information Processing Society. 90

[Andreasen *et al.*, 2003c] Troels Andreasen, Henrik Bulskov, and Rasmus Knappe. Similarity graphs. In E. Suzuki, Shusaku Tsumoto, N. Zhong, and Z.W. Ras, editors, *14th International Symposium on Methodologies for Intelligent Systems*, pages 668–672, Maebashi, Japan, 2003. ISMIS 2003. International Symposium on Methodologies for Intellignet Systems. 50, 80

[Andreasen *et al.*, 2004] Troels Andreasen, Per Anker Jensen, Jørgen Fischer Nilsson, Patrizia Paggio, Bolette Sandford Pedersen, and Hanne Erdman Thomsen. Content-based text querying with ontological descriptors. *Data Knowl. Eng.*, 48(2):199–219, 2004. ISSN 0169-023X. 70

[Andreasen *et al.*, 2005a] Troels Andreasen, Henrik Bulskov, and Rasmus Knappe. On automatic modeling and use of domain-specific ontologies. In Mohand-Said Hacid, Neil V. Murray, Z.W. Ras, and Shusaku Tsumoto, editors, *15th International Symposium on Methodologies for Intelligent Systems*, Saratoga Springs, New York, 2005. ISMIS 2005. 80, 115, 137

[Andreasen *et al.*, 2005b] Troels Andreasen, Rasmus Knappe, and Henrik Bulskov. Domain-specific similarity and retrieval. In Yingming Liu, Guoqing Chen, and Mingsheng Ying, editors, *11th International Fuzzy Systems Association World Congress*, volume 1, pages 496–502, Beijing, China, 2005. IFSA 2005, Tsinghua University Press. 107, 128, 129

[Andreasen, 2001] Troels Andreasen. Query evaluation based on domain-specific ontologies. In *Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS Int. Conf. (IFSA/NAFIPS'2001)*, volume 4, pages 1844–1849, Vancouver, BC, Canada, July 2001. 135

[Appelt and Israel, 1999] Douglas E. Appelt and David J. Israel. Introduction to information extraction technology: A tutorial prepared for ijcai-99. Available at http://www.ai.sri.com/userappelt/ie-tutorial/, 1999. 84

[Austin, 2001] Brice Austin. Mooers' law: in and out of context. *J. Am. Soc. Inf. Sci. Technol.*, 52(8):607–609, 2001. 9

[Baader and Nutt, 2003] Franz Baader and Werner Nutt. Basic description logics. In Baader et al. [2003], pages 43–95. 43

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications.* Cambridge University Press, 2003. 167, 176

[Baeza-Yates and Ribeiro-Neto, 1999] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999. 12, 14, 16, 18, 19, 28

[Bateman *et al.*, 1989] John Bateman, Robert Kasper, and Johanna Moore. A general organization of knowledge for natural language processing: The penman upper model. Technical report, Information Sciences Institute, Marina del Rey, California, 1989. 52

[Bateman, 1990] John A. Bateman. Upper modeling: organizing knowledge for natural language processing. In *5th. International Workshop on Natural Language Generation, 3-6 June 1990*, Pittsburgh, PA., 1990. 52

[Bechhofer *et al.*, 2004] DSean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *OWL Web Ontology Language Reference.* http://www.w3.org/TR/2004/REC-owl-ref-20040210/, 2004. 48, 61

[Beckwith *et al.*, 1993] Richard Beckwith, George A. Miller, and Randee Tengi. Design and implementation of the wordnet lexical database and searching software, 1993. 139

[Berners-Lee, 1998] Tim Berners-Lee. Semantic web roadmap, 1998. 2

[Bookstein, 1980] Abraham Bookstein. Fuzzy request: and approach to weighted boolean searches. *Journal of the American Society for Information Science*, 31:240–247, 1980. 20, 23

[Bordogna and Pasi, 2001] Gloria Bordogna and Gabriella Pasi. Modeling vagueness in information retrieval. *Lectures on information retrieval*, pages 207–241, 2001. 20

[Borgo *et al.*, 1996a] Stefano Borgo, Nicola Guarino, and Claudio Masolo. A pointless theory of space based on strong connection and congruence. In Luigia Carlucci Aiello, Jon Doyle, and Stuart Shapiro, editors, *KR'96: Principles of Knowledge Representation and Reasoning*, pages 220–229. Morgan Kaufmann, San Francisco, California, 1996. 51

[Borgo *et al.*, 1996b] Stefano Borgo, Nicola Guarino, and Claudio Masolo. Statified ontologies: the case of physical objects. In *Proceedings of of the ECAI-96 Workshop on Ontological Engineering*, pages 5–15, Budapest, 1996. ECAI. 51

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the kl-one knowledge representation system. *Cognitive Science: A Multidisciplinary Journal*, 9(2):171–216, 1985. 42

[Brachman *et al.*, 1985] Ronald J. Brachman, R. Fikes, and Hector J. Levesque. Krypton: A functional approach to knowledge representation. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 411–429. Kaufmann, Los Altos, CA, 1985. 42

[Brachman, 1985] Ronald J. Brachman. On the epistemological status of semantic networks. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 191–215. Kaufmann, Los Altos, CA, 1985. 40

[Brickley and Guha, 2004] Dan Brickley and R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema.* http://www.w3.org/TR/2004/REC-rdf-schema-20040210/, 2004. 47

[Brill, 1995] Eric Brill. Transformation-based error-driven learning amd natural language processing: a case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995. 71

[Brill, 1997] Eric Brill. A simple rule-based part of speech tagger, 1997. 150

[Brink *et al.*, 1994] Chris Brink, K. Britz, and R. A. Schmidt. Peirce algebras. *Formal Aspects of Computing*, 6(3):339–358, 1994. 44

[Brink, 1981] Chris Brink. Boolean algebra. *Journal of Algebra*, 71(2):291–313, August 1981. 45

[Budanitsky, 1999] A. Budanitsky. Lexical semantic relatedness and its application in natural language processing, 1999. 88

[Budanitsky, 2001] A. Budanitsky. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures, 2001. 91, 109

[Buell and Kraft, 1981] Duncan A. Buell and Donald H. Kraft. Performance measurement in a fuzzy retrieval environment. In *SIGIR '81: Proceedings of the 4th annual international ACM SIGIR conference on Information storage and retrieval*, pages 56–62, New York, NY, USA, 1981. ACM Press. 29

[Bulskov and Thomsen, 2005] Henrik Bulskov and Hanne Erdman Thomsen. Integration of a formal ontological framework with a linguistic ontology. In Bodil Nistrup Madsen and Hanne Erdman Thomsen, editors, *7th International Conference on Terminology and Knowledge Engineering*, pages 211–224, Copenhagen, Denmark, 2005. TKE 2005. 51, 59

[Bulskov *et al.*, 2002] Henrik Bulskov, Rasmus Knappe, and Troels Andreasen. On measuring similarity for conceptual querying. In *FQAS '02: Proceedings of the 5th International Conference on Flexible Query Answering Systems*, pages 100–111. Springer-Verlag, 2002. 91, 102

[Bulskov *et al.*, 2004] Henrik Bulskov, Rasmus Knappe, and Troels Andreasen. On querying ontologies and databases. In Henning Christiansen, Mohand-Said Hacid, Troels Andreasen, and Henrik Legind Larsen, editors, *LNAI 3055, 6th International Conference on Flexible Query Answering Systems*, Lyon, France, 2004. Flexible Query Answering Systems, Springer-Verlag. 134, 136

[Carlson and Nirenburg, 1990] L. Carlson and S. Nirenburg. World modeling for nlp. Technical report, Center for Machine Translation, Carnegie Mellon UniversityPittsburgh, PA, 1990. 52

[Charniak *et al.*, 1996] Eugene Charniak, Glenn Carroll, John Adcock, Anthony R. Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael L. Littman, and John McCann. Taggers for parsers. *Artificial Intelligence*, 85(1-2):45–57, 1996. 149

[Chen and Chen, 2005] Shi-Jay Chen and Shyi-Ming Chen. Fuzzy information retrieval based on geometric-mean averaging operators. *Computers & Mathematics with Applications*, 49(7-8):1213–1231, 2005. 22

[Chen, 1976] Peter Pin-Shan S. Chen. The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976. 46

[Cleverdon and Mills, 1997] Cyril W. Cleverdon and J. Mills. *The testing of index language devices*, pages 98–110. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1-55860-454-5. 65, 67

[Cohen and Kjeldsen, 1987] Paul R. Cohen and Rick Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage.*, 23(4):255–268, 1987. ISSN 0306-4573. 88

[Collins and Loftus, 1988] A. Collins and E. F. Loftus. A spreading-activation theory of semantic processing. In A. Collins and E. E. Smith, editors, *Readings in Cognitive Science: A Perspective from Psychology and Artificial Intelligence*, pages 126–136. Kaufmann, San Mateo, CA, 1988. 88, 125

[Connolly *et al.*, 2001] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. *DAML+OIL (March 2001) Reference Description.* http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218, 2001. 48

[Cooper, 1991] William S. Cooper. Some inconsistencies and misnomers in probabilistic information retrieval. In *SIGIR '91: Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–61, New York, NY, USA, 1991. ACM Press. 18, 19

[Cooper, 1997] William S. Cooper. Getting beyond boole. *Readings in information retrieval*, pages 265–267, 1997. 13, 14

[Corcho *et al.*, 2003] Oscar Corcho, Mariano Fernández-López, and Asuncón Gómez-Pérez. Methodologies, tools and languages for building ontologies: where is their meeting point? *Data Knowl. Eng.*, 46(1):41–64, 2003. 38, 47

[Crestani *et al.*, 1998] Fabio Crestani, Mounia Lalmas, Cornelis J. Van Rijsbergen, and Iain Campbell. Is this document relevant? probably: a survey of probabilistic models in information retrieval. *ACM Comput. Surv.*, 30(4):528–552, 1998. 10, 17

[Cross, 1994] Valerie Cross. Fuzzy information retrieval. *Journal of Intelligent Information Systems*, pages 29–56, 1994. 22, 24

[Davis *et al.*, 1993] Randall Davis, Howard E. Shrobe, and Peter Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993. 39, 64

[Edmundson and Wyllys, 1961] H. P. Edmundson and R. E. Wyllys. Automatic abstracting and indexing - survey and recommendations. *Commun. ACM*, 4(5):226–234, 1961. 66

[Escudero *et al.*, 2000] Gerard Escudero, Llu'is Màrquez, and German Rigau. Boosting applied to word sense disambiguation. In Ramon L'opez de M'antaras and Enric Plaza, editors, *Proceedings of ECML-00, 11th European Conference on Machine Learning*, pages 129–141, Barcelona, ES, 2000. Springer Verlag, Heidelberg, DE. 76

[Fellbaum, 1998] Christiane Fellbaum. Modifiers in wordnet. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 1, pages 23–46. MIT Press, 1998. 56

[Fensel *et al.*, 2000] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Stefan Decker, Michael Erdmann, and Michel C. A. Klein. Oil in a nutshell. In *Knowledge Acquisition, Modeling and Management*, pages 1–16, 2000. 48

[Francis and Kucera, 1964] W. N. Francis and H. Kucera. *BROWN CORPUS MAUNAL*. Department of Linguistics, Brown University, 1964. 97, 148

[Fuhr, 1992] Norbert Fuhr. Probabilistic models in information retrieval. *The Computer Journal*, 35(3):243–255, 1992. 17, 18

[Gangemi *et al.*, 2003] Aldo Gangemi, Nicola Guarino, Claudio Masolo, and Alessandro Oltramari. Sweetening wordnet with dolce. *AI Mag.*, 24(3):13–24, 2003. ISSN 0738-4602. 60, 145

[Genesereth, 1991] M. R. Genesereth. Knowledge interchange format. In J. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proc. of the Second International Conference (KR'91)*, pages 599–600. Kaufmann, San Mateo, CA, 1991. 46, 51

[Gómez-Pérez *et al.*, 2004] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological engineering*. Springer-Verlag, 2004. 34, 36, 46

[Goerz *et al.*, 2003] Günther Goerz, Kerstin Bücher, Bernd Ludwig, Frank-Peter Schweinberger, and Iman Thabet. Combining a lexical taxonomy with domain ontologies in the erlangen dialogue system. In B. C. Smith, editor, *KI - 2003 workshop 11 - Reference Ontologies vs. Applications Ontologies*, 2003. 59

[Gruber, 1992] Thomas Robert Gruber. Ontolingua: A mechanism to support portable ontologies, 1992. 46, 51

[Gruber, 1993] Thomas Robert Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, June 1993. 36

[Guarino and Giaretta, 1995] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases: towards a terminological clarification. In

N. Mars, editor, *Towards very large knowledge bases*, pages 25–32, Amsterdam (NL), April 1995. 2nd international conference on building and sharing very large-scale knowledge bases (KBKS), Ensche, IOS press. 32, 35, 36

[Guarino and Welty, 2004] Nicola Guarino and Christopher A. Welty. An overview of ontoclean. In *Handbook on Ontologies*, pages 151–172. 2004. 60

[Guarino, 1998] Nicola Guarino. Formal ontology and information systems. In Nicola Guarino, editor, *Formal Ontology in Information Systems*, pages 3–15, Trento, Italy, 1998. 1st International Conference on Formal Ontologies in Information Systems, FOIS'98, IOS press. 37

[Haav and Lubi, 2001] Hele-Mai Haav and Tanel-Lauri Lubi. A survey of concept-based information retrieval tools on the web. In *Proc. of 5th East-European Conference ADBIS 2001*, volume 2, pages 29–41, Vilnius Technika, 2001. 59

[Hayes, 1985] P. J. Hayes. Some problems and non-problems in representation theory. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 3–22. Kaufmann, Los Altos, CA, 1985. 42

[Heflin *et al.*, 1999] J. Heflin, Jim Hendler, and S. Luke. Shoe: A knowledge representation language for internet applications. Technical report, Dept. of Computer Science, University of Maryland at College Park, 1999. 47

[Hirst and St-Onge, 1998] Graeme Hirst and David St-Onge. Lexical chains as representation of context for the detection and correction malapropisms. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database.* MIT Press, 1998. 101

[Horrocks, 1998] Ian Horrocks. The fact system. In de Swart, editor, *Automated Reasoning with Analytic Tableaux and Related Methods: International Conference Tableaux'98*, number 1397 in Lecture Notes in Artificial Intelligence, pages 307–312. Springer-Verlag, May 1998. ISBN 3-540-64406-7. 48, 61

[Ingwersen, 1992] Peter Ingwersen. *Information retrieval interaction.* Taylor Graham Publishing, London, UK, UK, 1992. 6, 27

[J. Jiang, 1997] D. Conrath J. Jiang. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings on International Conference on Research in Computational Linguistics, Taiwan*, pages 19–33, 1997. 98, 99

[Jastrezembski, 1981] J. E. Jastrezembski. Multiple meanings, number of relationed meanings, frequency of occurrence, and the lexicon. *Cognitive Psychology*, 13:278–305, 1981. 58

[Jones and Willett, 1997] Karen Sparck Jones and Peter Willett. Overall introduction. *Readings in information retrieval*, pages 1–7, 1997. 7, 9

[Jones, 1972] Karen Sparck Jones. Exhaustivity and specificity. *Journal of Documentation*, 60(5):493–502, 1972. 65

[Jurafsky and Martin, 2000] Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2000. ISBN 0130950696. 69, 70

[Kifer *et al.*, 1995] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of ACM*, 42:741–843, July 1995. 47

[Klir and Yuan, 1995] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice Hall, 1995. 21, 22

[Knappe *et al.*, 2005] Rasmus Knappe, Henrik Bulskov, and Troels Andreasen. Perspectives on ontology-based querying. *International Journal of Intelligent Systems*, 2005. to appear. 104

[Knappe, 2006] Rasmus Knappe. *Measures of Semantic Similarity and Relatedness for Use in Ontology-based Information Retrieval.* PhD thesis, Roskilde University, 2006. 108

[Knight and Luk, 1994] Kevin Knight and Steve K. Luk. Building a large-scale knowledge base for machine translation. In *AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1)*, pages 773–778, Menlo Park, CA, USA, 1994. American Association for Artificial Intelligence. ISBN 0-262-61102-3. 52

[Kraaij, 2004] Wessel Kraaij. *Variations on Language Modeling for Information Retrieval.* PhD thesis, University of Twente, June 2004. 15

[Lancaster, 1979] Frederick Wilfrid Lancaster. *Information Retrieval Systems: Characteristics Testing, Evaluation.* John Wiley and Sons, second edition, 1979. 6, 7, 9

[Landes *et al.*, 1998] Shiri Landes, Claudia Leacock, and Randee Tengi. *Building semantic concordances*, chapter 8. MIT Pres, 1998. 57

[Lassila and McGuinness, 2001] Ora Lassila and Deborah McGuinness. The role of frame-based representation on the semantic web. Technical report kls-01-02, Knowledge Systems Laboratory, Standford University, Standford, California, 2001. 37

[Lassila and Swick, 1999] Ora Lassila and Ralph R. Swick. *Resource Description Framework (RDF) Model and Syntax Specification.* http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, 1999. 47

[Leacock and Chodorow, 1998] Claudia Leacock and Martin Chodorow. Combining local context and wordnet similarity for word sense identification. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database.* MIT Press, 1998. 96

[Levesque and Brachman, 1985] Hector J. Levesque and Ronald J. Brachman. A fundamental tradeoff in knowledge representation and reasoning (revised version). In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 41–70. Kaufmann, Los Altos, CA, 1985. 41

[Lin, 1997] Dekang Lin. Using syntactic dependency as local context to resolve word sense ambiguity. In *ACL*, pages 64–71, 1997. 89, 99

[Lin, 1998] Dekang Lin. An information-theoretic definition of similarity. In Jude W. Shavlik, editor, *ICML*, pages 296–304. Morgan Kaufmann, 1998. ISBN 1-55860-556-8. 89, 99

[Lucarella, 1990] Dario Lucarella. Uncertainty in information retrieval: An approach based on fuzzy sets. In *Proceedings of the International Conference on Computers and Communications*, pages 809–814, 1990. 20, 23

[Luhn, 1958] H. P. Luhn. The automatic creation of literature abstracts. In *IBM Journal of Research and Development*, volume 2, pages 159–165, 1958. 11

[MacGregor, 1991] Robert M. MacGregor. Inside the loom description classifier. *SIGART Bull.*, 2(3):88–92, 1991. 47

[Madsen *et al.*, 2001] Bodil Nistrup Madsen, Bolette Sandford Pedersen, and Hanne Erdman Thomsen. Defining semantic relations for ontoquery. In Per Anker Jensen and Peter Skadhauge, editors, *Ontologies and Lexical Knowledge Bases, 1st International Workshop, OntoLex 2000*, Proceedings of the First International OntoQuey Workshop, pages 57–88. University of Southern Denamrk, 2001. 59

[Manning and Schütze, 1999] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, 1999. 150

[Maron and Kuhns, 1997] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Readings in information retrieval*, pages 39–46, 1997. 16

[Masters, 1993] Timothy Masters. *Practical neural network recipes in C++*. Academic Press Professional, Inc., San Diego, CA, USA, 1993. 153

[McDermott, 1976] Drew McDermott. Artificial intelligence meets natural stupidity. *SIGART Bull.*, 57:4–9, 1976. 42

[Mihalcea and Moldovan, 2001] Rada Mihalcea and Dan Moldovan. Automatic generation of a coarse grained wordnet. In *NAACL Workshop on WordNet and Other Lexical Resources*, pages 35–41, Pittsburgh, PA, 2001. 77

[Miller and Charles, 1991] George A. Miller and W. G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28, 1991. 108

[Miller *et al.*, 1990] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and K.J. Miller. Introduction to wordnet: an on-line lexical database. *International Journal of Lexicography*, 3(4):235 – 244, 1990. 50, 52, 53

[Miller *et al.*, 1994] George A. Miller, Martin Chodorow, Shiri Landes, Claudia Leacock, and Robert G. Thomas. Using a semantic concordance for sense identification. In *Proc. of the ARPA Human Language Technology Workshop*, pages 240–243, 1994. 58, 76, 142, 148

[Miller, 1990] George A. Miller. Wordnet: an online lexical database. *International Journal of Lexicography*, 3(4), 1990. 97

[Miller, 1995] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995. 50, 53, 54

[Miller, 1998] George A. Miller. Nouns in wordnet. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 1, pages 23–46. MIT Press, 1998. 55

[Minsky, 1985] M. Minsky. A framework for representing knowledge. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 245–262. Kaufmann, Los Altos, CA, 1985. 40

[Motro, 1990] A. Motro. Imprecision and incompleteness in relational databases: survey. *Inf. Softw. Technol.*, 32(9):579–588, 1990. 20

[Nancy and Véronis, 1998] Ide Nancy and Jean Véronis. Word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1–40, 1998. 73, 74, 75, 76

[Nardi and Brachman, 2003] Daniele Nardi and Ronald J. Brachman. An introduction to description logics. In Baader et al. [2003], pages 1–40. 42

[Ng, 1999] Kenny Ng. A maximum likelihood ratio information retrieval model. In *Eighth Text REtrieval Conference (TREC-8)*, 1999. 19

[Niles and Pease, 2003] Ian Niles and Adam Pease. Linking lexicons and ontologies: Mapping wordnet to the suggested upper merged ontology. In *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE03)*, Las Vegas, Nevada, 2003. 51

[Niles and Terry, 2004] Ian Niles and Allan Terry. The milo: A general-purpose, mid-level ontology. *Information and Knowledge Engineering*, 2004. 52

[Nilsson, 2001] Jørgen Fischer Nilsson. A logico-algebraic framework for ontologies – ontolog. In Per Anker Jensen and Peter Skadhauge, editors, *Proceedings of the First International OntoQuery Workshop – Ontology-based interpretation of NP's, Department of Business Communication and Information Science*, Kolding, Denmark, 2001. University of Southern Denmark. 49

[NLG, 2006] NLG. The natural language group. http://www.isi.edu/natural-language/people/hovy.html, 2006. 52

[Object Management Group,] Inc. Object Management Group. Uml - resource page. Object Management Group webside: http://www.uml.org/. 46

[OntoQuery, 2005] OntoQuery. The ontoquery project net site: www.ontoquery.dk, 2005. 5, 70

[Osgood, 1952] Charles E. Osgood. The nature and measurement of meaning. *Psychological Bulletin*, 49(3):204, 1952. 88

[Paoli *et al.*, 2004] Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, and François Yergeau. *Extensible Markup Language (XML) 1.0 (Third Edition)*. http://www.w3.org/TR/2004/REC-xml-20040204/, 2004. 47

[Pease and Carrico, 1997] Adam Pease and T. Carrico. The jtf atd core plan representation: A progress report. In *Proceedings of the AAAI Spring Symposium on*, Stanford, CA, 1997. 51

[Pease *et al.*, 2001] Adam Pease, Ian Niles, and Teknowledge Corporation. Towards a standard upper ontology, 2001. 51

[Pedersen and Keson, 1999] Bolette Sandford Pedersen and Britt Keson. Simple - semantic information for multifunctional plurilingual lexica: Some danish examples on concrete nouns. In *SIGLEX99: Standardizing Lexical Resources, Association of Computational Linguistics*. ACL99 Workshop, Maryland, 1999. 51

[Pedersen and Nimb, 2000] Bolette Sandford Pedersen and Sanni Nimb. Semantic encoding of danish verbs in simple adapting a verb-framed model to a satellite-framed language. In *Proceedings from 2nd Internal Conference on Language Resources and Evaluation*, pages 1405–1412, Athens, Greece, 2000. 51

[Pustejovsky, 1991] James Pustejovsky. The generative lexicon. *Comput. Linguist.*, 17(4):409–441, 1991. 48

[Quillian, 1968] M. Quillian. *Semantic Memory*, chapter ?, pages 216–270. MIT Press, 1968. Semantic Information Processing. 88

[Quillian, 1985] M. R. Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 97–118. Kaufmann, Los Altos, CA, 1985. 40

[Quirk *et al.*, 1985a] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and J. Svartvick. *A Comprehensive Grammar of the English Language.* Longman, 1985. 79

[Quirk *et al.*, 1985b] Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. *A Comprehensive Grammar of the English Language.* Longman, New York, 1985. iindexQuirk, R.iindexGreenbaum, S.iindexLeech, G.iindexSvartvik, J. 151

[Rada and Bicknell, 1989] Roy Rada and Ellen Bicknell. Ranking documents with a thesaurus. *JASIS*, 40(5):304–310, 1989. 88

[Rada *et al.*, 1989] Roy Rada, H. Mili, Ellen Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, January 1989. 91, 96

[Resnik, 1995] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, pages 448–453, 1995. 76, 89, 115, 163

[Resnik, 1999] Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language, 1999. 96, 97, 115

[Robertson and Jones, 1976] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976. 16, 18

[Robertson and Walker, 1997] Stephen E. Robertson and S. Walker. Some simple effective approximations to the 2poisson model for probabilistic weighted retrieval. *Readings in information retrieval*, pages 345–354, 1997. 19

[Robertson, 1997] Stephen E. Robertson. The probability ranking principle in ir. *Readings in information retrieval*, pages 281–286, 1997. 10, 17

[Rubinstein and Goodenough, 1965] H. Rubinstein and J. B. Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10), 1965. 108

[Ruimy *et al.*, 1998] Nilda Ruimy, Ornella Corazzari, Elisabetta Gola, Antonietta Spanu, Nicoletta Calzolari, and Antonio Zampolli. Le-parole project: The italian syntactic lexicon. In *EURALEX'98*, volume 1, page 259, Université de Liège, 1998. 71

[Salton and Buckley, 1997] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Readings in information retrieval*, pages 323–328, 1997. 67

[Salton and Lesk, 1997] Gerard Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Readings in information retrieval*, pages 60–84, 1997. 15

[Salton and McGill, 1986] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. 12, 13, 15, 16

[Salton and McGill, 1997] Gerard Salton and Michael J. McGill. The smart and sire experimental retrieval systems. *Readings in information retrieval*, pages 381–399, 1997. 10

[Salton *et al.*, 1983] Gerard Salton, Edward A. Fox, and Harry Wu. Extended boolean information retrieval. *Commun. ACM*, 26(11):1022–1036, 1983. ISSN 0001-0782. 25

[Schmidt-Schaubss and Smolka, 1991] Manfred Schmidt-Schaubss and Gert Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, 1991. 44

[Shannon and Weaver, 1949] C. E. Shannon and W. Weaver. *A Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois, 1949. 97, 100

[Smith, 1990] Maria Elena Smith. *Aspects of the P-Norm model of information retrieval: syntactic query generation, efficiency, and theoretical properties.* PhD thesis, Cornell University, Ithaca, NY, Ithaca, NY, USA, 1990. 25

[Sowa, 2000] John F. Sowa. *Knowledge representation: logical, philosophical and computational foundations.* Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2000. 33, 34, 35

[St-Onge, 1995] David St-Onge. Detecting and correcting malapropisms with lexical chains, 1995. 101

[SUMO, 2006] SUMO. Standard upper ontology working group (suo wg). http://suo.ieee.org/, 2006. 51

[Sussna, 1993] Michael Sussna. Word sense disambiguation for tree-text indexing using a massive semantic network. In Bharat Bhargava, Timothy Finin, and Yelena Yesha, editors, *Proceedings of the 2nd International Conference on Information and Knowledge Management*, pages 67–74, New York, NY, USA, November 1993. ACM Press. 94, 95

[Tomuro, 2001] Noriko Tomuro. Tree-cut and a lexicon based on systematic polysemy. In *NAACL*, 2001. 77

[Tversky, 1977] Amos Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, July 1977. 90

[Uschold and Jasper, 1999] Mike Uschold and Robert Jasper. A framework for understanding and classifying ontology applications. In *IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5)*, 1999. 46

[Van Rijsbergen, 1979] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition.* Dept. of Computer Science, University of Glasgow, 1979. 6, 17

[Van Rijsbergen, 1986] C. J. Van Rijsbergen. A non-classical logic for information retrieval. *Comput. J.*, 29(6):481–485, 1986. 19

[Voorhees, 1998] Ellen M. Voorhees. *Using WordNet for Text Retrieval*, chapter 12. MIT Press, 1998. 75

[Waller and Kraft, 1979] W. G. Waller and Donald H. Kraft. A mathematical model of a weighted boolean retrieval system. *Inf. Process. Manage.*, 15(5):235–245, 1979. 25

[Webster, 1993] M. Webster. Webster's third new international dictionary, 1993. 32

[Woods, 1985] W. A. Woods. What's in a link: Foundations for semantic networks. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, pages 217–241. Kaufmann, Los Altos, CA, 1985. 40, 42

[WordNet, 2005] WordNet. Wordnet 2.1 reference manual. http://wordnet.princeton.edu/man/, 2005. Cognitive Science Laboratory, Princeton University. 54

[Wu and Palmer, 1994] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, Morristown, NJ, USA, 1994. Association for Computational Linguistics. 95

[Yager, 1987] Ronald R. Yager. A note on weighted queries in information retrieval systems. *Journal Of The American Society For Information Science*, 38(1):23–24, 1987. 20

[Yager, 1988] Ronald R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-18(1):183–190, January 1988. 25

[Yager, 2000] Ronald R. Yager. A hierarchical document retrieval language. *Information Retrieval*, 3(4):357–377, 2000. 25, 26, 129, 135

[Zadeh, 1993] Lotfi A. Zadeh. Fuzzy sets. In *Readings in Fuzzy Sets for Intelligent Systems*. Morgan Kaufmann Publishers Inc., 1993. 20