CONCEPT-BASED SEARCH USING PARALLEL QUERY EXPANSION

by

RAHUL RAJIV JOSHI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2006

To my family and all those who have supported me.

ACKNOWLEDGMENTS

I would like to thank my supervising professor, Dr. Yuksel Alp Aslandogan, for constantly motivating and encouraging me, and also for his invaluable advice during the course of my master's studies, and this work in particular.

I wish to thank Dr. Ramez A. Elmasri and Dr. Gautam Das for their interest in my research and for taking time to serve in my thesis committee.

The feedback of users in the evaluation of the proposed approach is greatly appreciated.

I am also extremely grateful to my parents and my sister for their support and encouragement. I also thank several of my relatives and friends who have helped me with guidance and suggestions throughout my career.

April 12, 2006

ABSTRACT

CONCEPT-BASED SEARCH USING PARALLEL QUERY EXPANSION

Publication No. ______

RAHUL RAJIV JOSHI, M.S.

The University of Texas at Arlington, 2006

Supervising Professor: Yuksel Alp Aslandogan

We address the problem of irrelevant results for short queries on Web search engines. Short queries fail to provide sufficient context to disambiguate possible meanings associated with the search terms resulting in a set of irrelevant pages that the user has to filter through navigation and sometimes examination.

First, we predict the potential concept topics, which are the domains for the search terms. This prediction is based on word occurrences and relationships observed in the various domains (categories) of a corpus. Next, we expand the search terms in each of the predicted domains in parallel. We then submit separate queries, specialized for each domain, to a general purpose search engine. The user is presented with categorized search results under the predicted domains.

The theoretical foundations of our approach include concept identification in the form of associated terms through latent semantic indexing, in particular the WordSpace model, one sense per collocation and one domain per discourse assumptions, and sense disambiguation through sufficient context.

User evaluations of our approach indicate that it helps the users avoid having to examine irrelevant Web search results, especially with shorter queries.

Another contribution of our work is the development of a Web-based corpus of documents including sufficiently rich collections in multiple subject categories. We also created a mapping between these subject categories from the Open Directory Project and the domains from WordNet Domains.

TABLE OF CONTENTS

Appendix

LIST OF FIGURES

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

The growth of the Web has made search engines the dashboards for our explorative journeys through the ever-increasing, accessible information. Search engines have grown up too— becoming faster and more accurate with multiple, advanced search features. However, they have not been able to fully conquer the complexity and solve the ambiguity of natural languages. There still exists a gap between the user's goals and the search engine's offerings. An approach to identify the possible concept areas that the user might be interested in and then searching for documents containing the relevant words is one of the possible solutions to bridge this gap.

1.1 Types of Web Search Engines

Search engines have evolved into different types [3], each catering to a particular search need. Some are general purpose (e.g., Google [4]), some target shoppers (e.g., BizRate [5]), some are tailored for kids (e.g., KidsClick [6]), while some others let users search multiple search engines (e.g., Hotbot [7]). There are directories (e.g., Open Directory Project [8]) which are Web page databases compiled by human editors. CiteSeer [9] is a service to search computer science documents, citations, and acknowledgments. There are also search engines like Ask.com [10] that advise the user to "narrow" or "expand" the search using some relevant words or phrases. The smart capabilites provided by such a search engine are most important to a user exploring a new area where some directions from the search engine could help.

1.2 Search Queries and Ambiguity

Query length is the number of terms in the query submitted to a search engine. Although query length for Web search engines is increasing [11], the average query length was 1.6 terms for popular queries and 2.2 terms over all queries according to a study [12]. The study analyzed hundreds of millions of queries submitted by approximately 50 million users to a major search engine over a seven day period. A word in a natural language can have different meanings, a characteristic called *polysemy.* According to the statistics provided by WordNet [13], the average poylsemy for words, excluding monosemous words, is 2.77 for nouns, 3.56 for verbs, 2.74 for adjectives, and 2.49 for adverbs. Moreover, there are many more words and senses that are not captured in WordNet yet. It is reasonable to say that there is a high probability to encounter queries that are ambiguous due to the many possible senses that a word can have. Thus, there is a potential to improve the user's search experience by query disambiguation. Query expansion can be employed to disambiguate the query by adding new terms, which are related to the original query in a particular context. Term co-occurrence can be used as a means to identify the related terms.

1.3 Concept and Concept Topics

When a user searches for a term, there is more to the "query" than what is actually entered. Humans think in terms of *concepts* but the search is performed using words. Many times the query terms are ambiguous words, unable to fully represent the concept that the user has in mind. The intended meaning of such words is described by other words commonly occuring in the vicinity or context of these words.

Table 1.1 shows some examples of words and their concepts. We observe that if considered alone, each word has no clear meaning. On the other hand, the word supplemented with some relevant words from a context together have an obvious meaning. In this thesis, we consider *concept* as a group of words that together identify a clear meaning of the intended context. The expanded query that our approach constructs, represents the concept for a given subject area. We define *concept topic* as the generalized subject area to which the concept belongs. We present two concept topic types: domains and categories. Domains are obtained from WordNet Domains [14] while categories are the categories from our corpora.

Table 1.1 Examples of Word, Concept, and Concept Topics

Word	**Concept**	**Concept Topic**	
		Domain	**Category**
cell	stem cell	*biology*	*Science_Biology*
circuit	circuit court	*law*	*Society_Law*
java	hazelnut java	*food*	*Shopping_Food*

1.4 System Goal, Design, and Methodology

The goal of this research is to study the effectiveness of using query expansion within a category or domain topic as applied to Web search. Latent semantic indexing (LSI) [2] has been used with query expansion to improve the precision of search on an intranet [15]. We aim to enhance and extend this idea to Web search.

We first train categorized WordSpace [16] models from text corpora and use these along with lexical resources viz., WordNet and WordNet Domains to obtain various senses and domains of each word. For each of these words, we obtain the related words and store in an embedded database. When a user query is received, the words and their related terms are looked-up and the expanded query is passed on to the search engine. The high-level design of the system is shown in Figure 1.1. The Web poses a greater

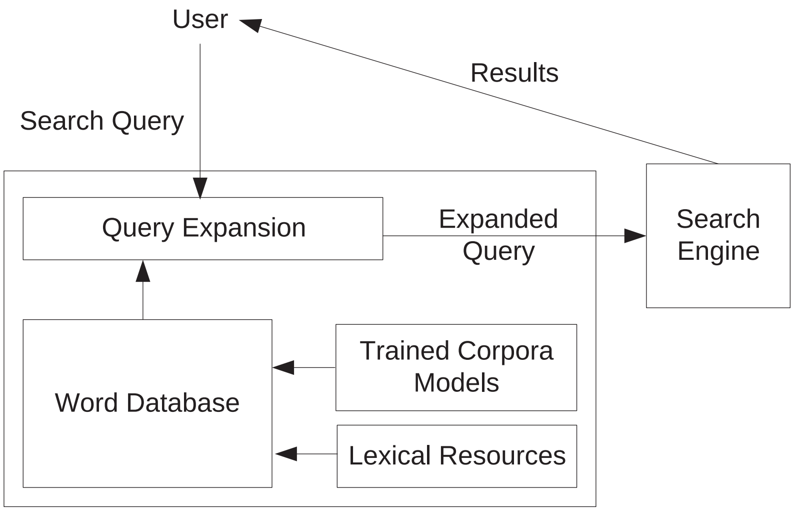


Figure 1.1 High level design of the system.

challenge in gathering training corpus to obtain contextual information. We prepared a text collection by downloading categorized Web pages from the Open Directory Project listing. An attempt was made to make the collection as broad as possible so as to cover as many categories as possible. The original hierarchical structure of the collection was preserved for up to the required levels. In order, to evaluate our results using available benchmark test collections, two other collections were used.

We used the Infomap NLP Software [17] package to train our text corpora and identify relevant terms for query expansion. Infomap uses a variant of Latent Semantic Indexing technique. Lexical information like word senses and word domains was obtained from WordNet and WordNet Domains [14] respectively. This was used to enhance the corpus categories by mapping the domains from WordNet Domains to our corpus categories.

Evaluation of the system was done by users for top ten results obtained from the Google Search engine for the expanded queries. Precision was calculated based on the relevance feedback provided by the users.

1.5 Breakdown of Chapters

The remainder of the thesis is organized as follows:

Chapter 2 surveys the related work in the areas of information retrieval, natural language processing, search engines, and query expansion.

Chapter 3 describes the construction of our Web directory corpus, the benchmark corpora we used, and the generation of Domain-Category mapping.

Chapter 4 describes the system architecture and the components of the system and also the implementation of these modules.

Chapter 5 describes the evaluation process and presents the results.

Chapter 6 concludes the thesis and provides a summary of our contribution.

CHAPTER 2

RELATED WORK

Although the area of Information Retrieval (IR) has been a part of computer science for many decades, it was directly used by only a fraction of the public. This changed with the advent of the World Wide Web and its mass reach and popularity. Unprecedented amount of information became available to the Web users in an easy-to-access manner. The need for the ability to find the relevant information in an equally easy way brought IR and Web search engines into the focus. In this chapter, we look at the research in the area of IR and search technology.

2.1 Overview of an IR System

A high level overview of an IR system is given in [1]. We briefly describe the system and then introduce the constituents of this system. Figure 2.1 presents an overview of a typical IR system. The user's query is input to the *query processing* module. This module can transform the query into a format more suitable and more effective for the retrieval process. The *retrieval model* finds the documents from the *index* which are

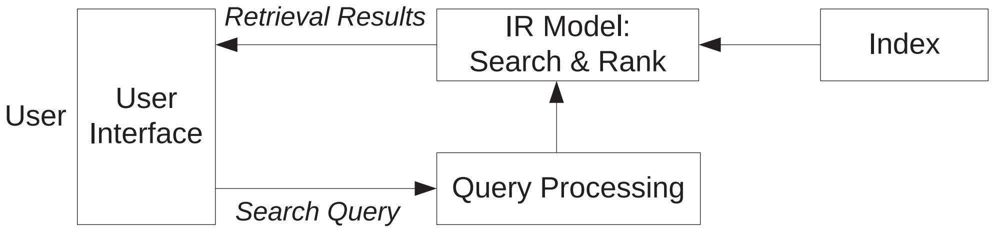


Figure 2.1 Overview of an IR system, adapted from [1].

relevant to the input query. The *index* itself consists of representations of the original documents in a convenient and quickly accessible form.

The various challenges in Web IR and the structure and working of a Web search engine are described in [18]. PageRank, a Web page ranking algorithm by them is presented in [19].

In this thesis, we focus on *query transformation* by pre-processing the query in various subject areas, and in parallel, to enhance the Web IR.

2.2 Transforming the Query by Query Expansion

When a user begins searching for some information, if the query entered is short, then there is a possibility that it is ambiguous due to the inherent nature of natural language. This is especially true for users exploring and searching for information in areas they do not have much expertise. In Query Expansion (QE), the initial query is appended with related, contextual, or synonymous terms so as to make the new query more complete to define the required concept area. QE has been shown to enhance the effectiveness of search on both the internet [20] as well as an intranet [15].

The term appending can be done either before the query is submitted, after the search is processed, or at both the stages. In our approach, we expand the query for each concept topic in parallel after retrieving the concept topics for the initial query. We then search again with the re-formulated query.

There are three types of QE techniques, viz., manual QE, automatic QE, and interactive QE [21]. In manual QE, the user manually modifies the initial query depending on the outcome of the initial search. In automatic QE, the system decides which terms to add in the reformulated query. Finally, in the interactive QE process, the system and the user together decide the new query. Our search approach in this thesis consists of interactive subject area selection, in that the user selects the concept topic, but the actual process of adding terms and creating the concept is automatic.

2.3 Information Retrieval Models

There are three classic IR models [22], viz., the set theoretic boolean model, the vector model, and the statistical probabilistic model. The generalized vector space model, latent semantic indexing [2] and WordSpace [16] models are alternate algebraic models. In this thesis, we use the WordSpace model which is thoroughly described in the next section.

2.4 Vectors and the WordSpace

We briefly describe the vector space and the latent semantic indexing models before presenting the details of WordSpace.

2.4.1 Vector Space Model

In vector space model [23], both documents and terms (words or phrases) are represented as vectors in an n-dimensional space where each dimension corresponds to a word. The relevant documents of a query are the documents having their vector representations closest to the query vector. In other words, the frequency of the occurrence of query terms in a document will determine the association between the query and the document. Generally the vectors are normalized and the similarity between vectors, calculated as the cosine of the angle between two vectors, is a simple dot product.

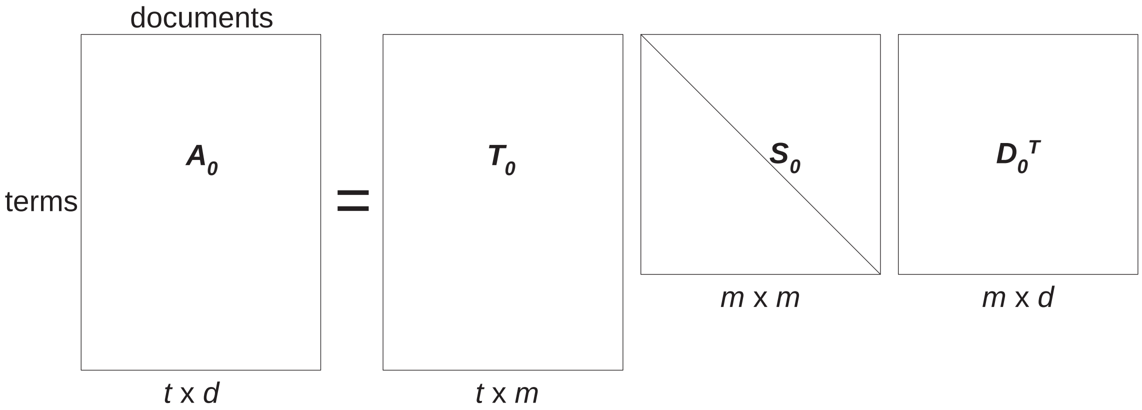


Figure 2.2 Singular value decomposition schematic for LSI, adapted from [2].

2.4.2 Latent Semantic Indexing (LSI)

In latent semantic indexing (LSI) [2], a term-by-document matrix is analyzed using Singular Value Decomposition (SVD). Using SVD, a $t \times d$ rectangular term document matrix A_0 is decomposed as follows:

$$A_0 = T_0 S_0 D_0^T$$

where the $t \times m$ matrix T_0 represents left singular vectors, the $m \times m$ matrix S_0 represents singular values, and the $m \times d$ matrix D_0^T represents right singular vectors. This is a rank-m model. A schematic (adapted from [2]) is shown in Figure 2.2. If we take the first k largest values of S_0, delete the rows and columns containing remaining smaller values, and also delete the corresponding columns of T_0 and D_0, we obtain a rank-k model with reduced dimensions:

$$A = TSD^T$$

where T is a $t \times k$ matrix, S is a $k \times k$ matrix, and D^T is a $k \times d$ matrix. D_0^T represents right singular vectors. This is a rank-k model and in the latent semantic space. Also,

$$A_0 \approx A$$

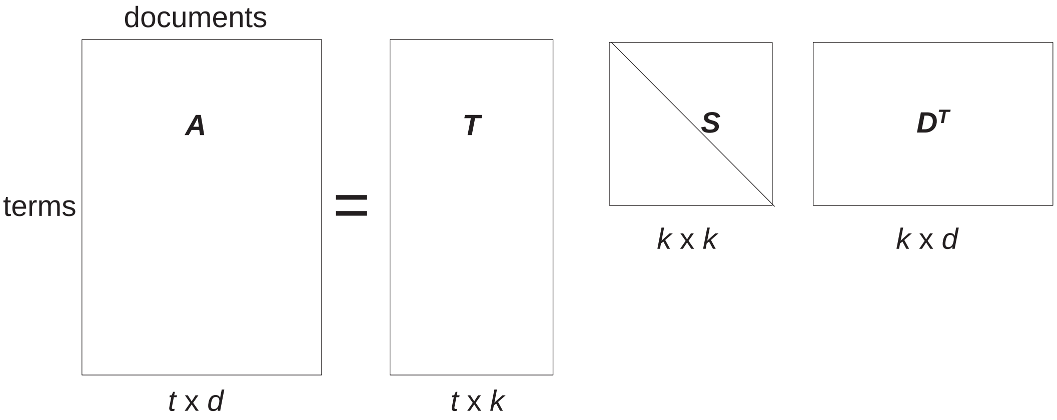


Figure 2.3 Schematic of reduction of the matrix for LSI, adapted from [2].

The reduction is explained in the Figure 2.3. In this reduced dimension space, only the strong associations are retained and the weak ones are ignored. Depending on the associations, the vectors for a document and those for the terms that do not occur in that document may be placed close in the space reduced space. The same is true for the vectors of documents and also for the vectors of terms.

By using this scheme in IR methods, we can overcome some problems that exist in the term-based IR methods using the standard indexing. The synonymy problem is solved because in LSI, the exact query term does not have to be present in a document for the document to be considered relevant. Although LSI is not very successful in dealing with the polysemy problem, it does attempt to get the *latent* meaning of the term.

2.4.3 WordSpace: The Word Vector Space

The word vector space or WordSpace model [16] uses a variant of LSI. It can represent words using word vectors, contexts using context vectors, and senses using sense vectors. In this thesis, we use the word vectors from the WordSpace model to find similarities between words.

The standard term-document matrix approach is not very suitable for measuring the similarities between terms. This is so because documents do not have equal sizes across corpora [24]. Another reason is that synonymous words do not generally co-occur in a document. These words are close in meaning but when it comes to the statistical model, this is not reflected in the simple term-document matrix. The co-occurrence of words (terms) in a document is not the best way to measure the relatedness of words [24]. The solution presented by the WordSpace model is to see if words occur close together in text. Instead of a term-by-document matrix used in standard LSI model, the WordSpace model uses a term-by-"content-bearing term" matrix. Some *content-bearing* words are selected and the co-occurrence of these with all the words of the text collection is measured. The co-occurrence can be measured by the distance in the form of the number of intervening words between the content-bearing words and all the words in the text within the same document.

In the texts below, the content-bearing words are shown in **bold** font while the other words used for our example are in *italics*.

Text 1: The DFW International **airport** is a very busy *facility* in terms of *airplane* traffic. The **airport** is the main hub for a major *airline*. The **airport** serves millions of passengers every year. The **airport** also handles considerable *cargo* traffic.

Text 2: The various modes of **travel** include *road*, *rail*, *aviation*, and *ship*. *Aviation* is the fastest mode of **travel** when the distances are large. For **travel** within a large city, *subway* is a fast option.

The term co-occurrence is shown in Table 2.1. The latent semantic analysis of the matrix is performed in a way similar to the one explained in section 2.4.2. The schematic for the singular value decomposition of the word-by-content bearing words matrix is shown in Figure 2.4. The reduction is shown in Figure 2.5.

Table 2.1 Term-by-"Content-bearing Term" Table

	airport	**travel**
airline	1	0
airplane	1	0
aviation	0	2
cargo	1	0
facility	1	0
rail	0	1
road	0	1
ship	0	1
subway	0	1

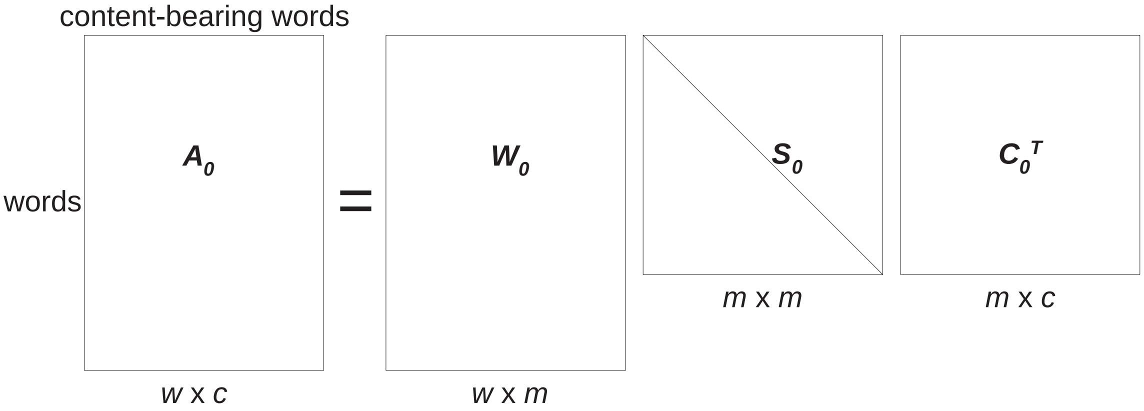


Figure 2.4 Singular value decomposition schematic for LSI in WordSpace, adapted from [2].

The Infomap-NLP software [17] implements the WordSpace model for text document corpus. It first removes *stop words*, which are common words like *if*, *a*, *the*, etc. with low information content, and then selects the set of c most frequent words as the content-bearing words. The value of c can be specified by the developer. In this thesis, we used this software to create models from our corpora and compute the similarity between words with c value of 1000.

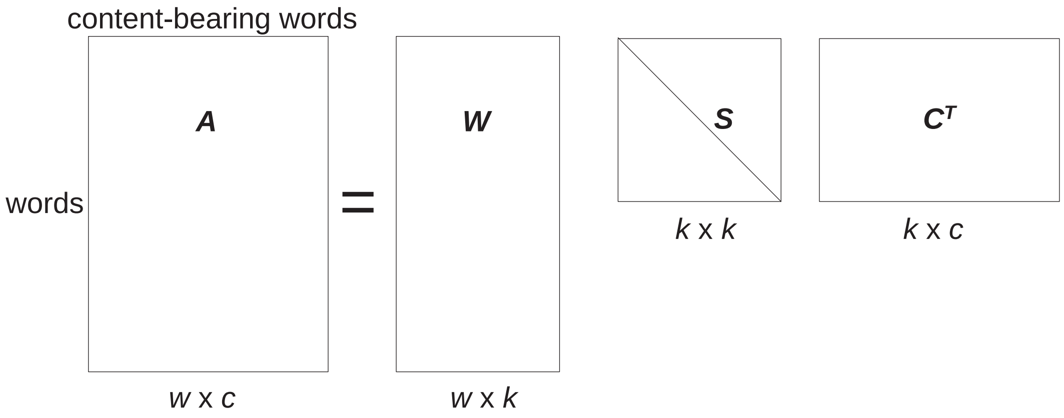


Figure 2.5 Schematic of reduction of the matrix for LSI in WordSpace, adapted from [2].

2.5 Test Collections for IR

To enable researchers to compare different techniques in IR, standard test collections or corpora are used. Some of the common ones are TREC [25], Reuters Corpus Volume 1 (RCV1) [26], BioMed Central's Corpus [27], etc. These can be considered as benchmark collections because the experimental results for these are available. The Web offers great potential as a source for obtaining text documents for IR experiments. Web directories like the Open Directory Project [8] provide an excellent listing of categorized lists of Web pages. An example of such a test collection is the TechTC-300 test collection [28] for text categorization.

2.6 Lexical Resources

Lexical resources can be used to supplement IR techniques in improving the effectiveness of search. Word sense disambiguation (WSD) is an important application of these resources which are available as dictionaries or databases. In this section, we present an overview of WordNet and WordNet Domains.

2.6.1 WordNet

WordNet [13] is a manually compiled lexical database for the English language. It organizes nouns, adjectives, verbs, and adverbs into synonym sets such that each set represents one concept. As an example, let us consider the sysnset for the English noun *java*. The WordNet 2.0 output is shown in Table 2.2. WordNet has been used for natural language processing, IR, language research and teaching, etc.

Table 2.2 WordNet Synset Look-up for the Noun *java*

Sense Number	**Synonym Set**	**Hypernym Set**
1	Java	island
2	coffee, java	beverage, drink, drinkable, potable
3	Java	object-oriented programming language, object-oriented programing language

2.6.2 WordNet Domains

WordNet Domains [14] is a lexical database that augments WordNet by assigning a domain label to every synset in WordNet. The domains are manually organized in a hierarchical tree of domain labels. By providing a high level attribute to every synset, the domains complement the information that is already available in WordNet. Let us consider the example *java* again. Since the noun has three senses in WordNet, we have at least three corresponding domains assigned for this word. These are shown in Table 2.3. The row entries in this table correspond to the ones in Table 2.2.

2.7 Distributional Aspects of Language Ambiguity

The one sense per discourse (OSD) [29] hypothesis related to a discourse effect on the use of ambiguous words. If a polysemous word is used multiple times in a well-

Table 2.3 WordNet Domains 3.1 Look-up for Senses of the Noun *java*

Sense Number	**Hypernym Set**	**Domain Set**
1	island	*geography*
2	beverage, drink, drinkable, potable	*food*
3	object-oriented programming language, object-oriented programing language	*computer science*

written discourse, then there is a strong likelihood that they share the same sense. The experiments showed that this tendency was verified with a probability of 98%.

Later research on polysemous words also examined the distributional aspects of language ambiguity. The study [30] used several definitions of collocation like direct adjacency, previous or next word, part-of-speech, and syntactic relationships. Experiments showed that the probability for an ambiguous word to have one sense in a given collocation was 90-99%.

The application of one sense per collocation hypothesis for fine-grained word sense distinctions [31], like those in WordNet, was found to give precision of the tune of 70%, a sharp drop from results reported by [30]. One sense per collocation was found to be valid across corpora with variations across corpora, following genre and topic variations.

The role of domain information in WSD was investigated in [32]. The hypothesis was that different domain labels like *Medicine*, *Architecture*, *Sport*, etc. for different senses could help in the WSD process. Domain-based disambiguation was found to achieve a high degree of precision compared to other systems, when tested for Senseval 2 data. A One Domain per Discourse (ODD) hypothesis, claiming that “multiple uses of a word in a coherent portion of text tend to share the same domain” was proposed by [32]. This suggested that the domain of a text was strongly related to the senses of the words in the text. An experiment carried out using WordNet Domains as the domain source verified the ODD hypothesis that there are only a few relevant domains in a text.

In this thesis, we rely on the conclusions made by both the one sense per collocation and the one domain per discourse hypotheses. In particular, we assume that multiple uses of a word in a set of texts belonging to the same topical category share the same semantic meaning.

2.8 Web Search Engines and Concept-based Web IR

In this section we explore the features of some existing search engines including one that provides enhanced search features. We also look at some recent work based on concept-based information access.

2.8.1 Google

Google [4] is today one of the leading search engines in terms of popularity [3]. Google introduced PageRank [19], an algorithm for ranking web pages based on the link structure of the Web. A link from page A to page B is interpreted as a vote by page A for page B. The more the number of links to a page, the greater its value in PageRank computation. In addition, the importance of the linking page is also considered. This information is combined with sophisticated hypertext matching analysis to arrive at the final measure of a Web page's importance and relevance to a query. These matching techniques analyze the full content of the text including the structure of the page and the location of words and the content of the pages linking to it. These automated, complex, and comprehensive techniques make it difficult to tamper with the results.

In this thesis, we use Google as our underlying search engine to search the Web using the expanded search queries.

2.8.2 Ask.com

Ask.com is a Web search engine that aims at providing most relevant, subject-specific search results using ExpertRank (earlier called Teoma) algorithm [10]. It ranks which ranks Web pages based on subject-specific popularity. Ask provides the user with the ability to narrow the search, expand the search, or to search for related names. For example, if the user searches for the word *java*, Ask provides options as mentioned in Table 2.4. Our approach in this thesis presents the concept topics to which the user query may belong and then expands the query based on the associated words from our corpus to obtain the concept. We focus on providing as many concepts as possible without asking the user to select the expansion terms. Thus, Ask's approach involves interactive query expansion because the user selects the new search query while in our approach the user simply selects the topic.

Table 2.4 Some of Ask.com's Helpful Options for the Query *java*

Feature	**Displayed Query Options**
Narrow Your Search	Java Programming Java Download Free Java Download Free Java Applets Java Applet Java Virtual Machine Java Tutorial Learn Java Programming History of Java Java Games
Expand Your Search	HTML Sun Microsystems Indonesia Coffee Visual Basic

2.8.3 Northern Light

Northern Light's Enterprise search engine [33] uses the principle of organizing results into a set of relevant categories [34], which was used earlier by the Northern Light Technology Web search engine. The same algorithm was used for searching both the Web and premium content information. A result set of records is first obtained by searching the database for the input user search query. A list of candidate categories is compiled by identifying common characteristics of result records and grouping such records. Weights are assigned to the candidate categories using a function of the identified common characteristics of the category records. Result categories are obtained by selecting from these candidate categories. The categories are displayed to the user and enable the user to perform the search without having to examine long linear list of results.

Northern Light's method provides result classification in four separate domains: subject, type, source, and language. The subject taxonomy consisted of 16,000 subject nodes in 9 levels [35], providing subject-specific classification and description. The taxonomies were constructed by librarians using different existing taxonomies. Studies suggested that the classifiers used had 90-95

Our approach in this thesis also identifies the set of subject categories (concept topics) for the user query. But for each single word in our database, we store this category information prior to the search process, that is, during pre-processing stage. Moreover, we do not categorize the search results after they are obtained. We perform search using different expansion terms for each possible category. Thus at its core, our approach is different from that of Northern Light although the user interface presents possible categories for the user query.

2.8.4 LSI-based Query Expansion on an Intranet

An experiment [15] to apply automatic query expansion using related terms obtained from LSI was performed on a corporate intranet. The existing search engine of the intranet was used to actually perform the search for the expanded queries. A total of 6,500 documents from three sub-domains within Volvo companies were used for the LSI process. The final thesaurus which was generated consisted of over 17,000 terms and up to 20 related terms for each of these terms. For every term, the related terms used for query expansion were obtained by taking the terms with a cosine measure greater than 0.2 in the latent space.

The evaluation was performed by 55 users for five search tasks. The search tasks consisted of a question and the users were told to formulate queries. Only top six results obtained from the internal search engine were used for the evaluation and precision computation. The difference betweeen the average precision for the unexpanded (average precision = 0.464) and the expanded (average precision = 0.451) was small but favorable for the unexpanded queries. Deeper analysis showed that explorative queries benefited from query expansion while targeted queries suffered from query drift.

2.9 Concept-based Information Access

Concept-based information access [36, 37] has been examined as a technique to improve quality of search results in Web IR. Two approaches were examined: the first one identifies concepts based on WSD, and the second one is based on representing concepts by a set of associated words using domain-specific corpora.

In the WSD approach, concepts are identified using evidence combination of supervised and unsupervised WSD techniques. It resulted in significant improvement in short queries for short documents.

If we assume that only one sense of a word is relevant for a domain, then this sense represents a concept. In the second approach, for a given word, related words were obtained from models built in WordSpace. The word with the related words represented a concept. Web search was performed using the concept and the results were analyzed for relevance to the intended sense. The LSI-based query expansion approach produced encouraging results for experiments performed on text corpora in travel and educational domains.

In this work, we extend this approach to more domains and with some improvements like more granular corpora and examining the use of lexical resources to improve the effectiveness. Our basic assumption in using WordSpace in ambiguity resolution is that “sense distinctions are mirrored by topical distinctions”, as described in [16].

2.10 Summary

The technique of IR has come a long way since the time the vector space model was first introduced. IR as a process has become so widespread that today it is the most important means of finding information for all of us. Because of this, and the need to make it more effective, newer models like LSI and WordSpace were proposed. Specialized search engines for specific purposes and using complex algorithms were developed. The quest for identifying the most relevant results for the user has turned towards identifying the meaning of the user’s query. It is evident that concept-based information access is one area that deserves some attention. We evaluate such a concept-based approach using topic-specific datasets (corpora), lexical resources and query expansion with concept identification and ambiguity resolution as our areas of focus.

CHAPTER 3

CORPORA AND WORDNET DOMAIN MAPPING

In this thesis, we present search results to the user based on the concept topic that is selected for the search. It is obvious that we have to first identify the concept topics for the user's search query. We have to define what concept topics mean and what forms they can assume. Our concept topics are closely related to semantic topic categories and to semantic domains in the context of Web information. In order to make our approach identify and recognize such concept topics, we need representative documents with known topic affiliations. Since we intended to make our approach as broad as possible, we needed documents belonging to as many topics as possible, in fact, ideally all possible topics. Moreover, because the Web is so dynamic, the need was for documents from the Web so that we could capture the most recent state of the information available on the Web.

This chapter describes our methodology to construct the UTA-295 corpus, a collection of categorized documents from the Web. It begins by giving an overview of the Open Directory Project (ODP) and then details our approach in creating the corpus. It also gives numerical statistics of our corpus and an overview of the other corpora that we use. We then explain the way we map semantic domains to categories in our corpora.

3.1 UTA-295 ODP Corpus: Generating a Corpus using Web Directories

The motivation to generate a corpus was the need to have a corpus to match the latest state of the Web and to have categorized sets of data with the depth and breadth as per our needs. On the other hand, the drawback of using a non-standard collection is that results cannot be easily compared with those from other researchers. We overcome the problem by simultaneously using two benchmark collections to evaluate our approach. We call our corpus *UTA-295 ODP-based Web corpus* but for convenience we also refer to it as the *UTA-295 ODP corpus*, where *UTA* stands for The University of Texas at Arlington and *295* represents the number of categories in the corpus obtained using the *ODP* [8]. We also present the structure and the hierarchical details of our corpus. In previous research [28], a collection for text categorization was generated from the Web and evaluated, which gave us further encouragement to generate our corpus.

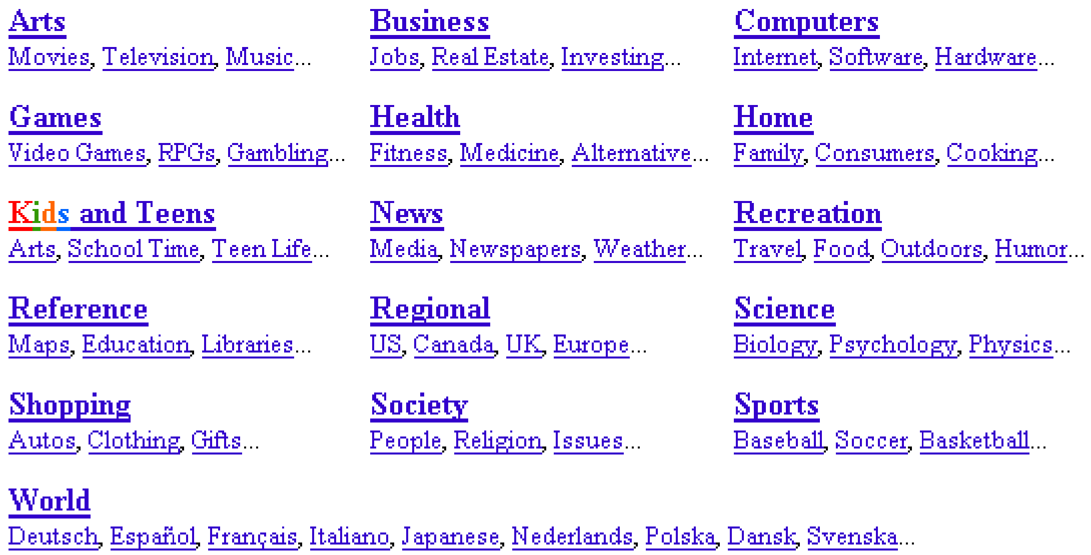
3.1.1 Overview of the ODP

The Open Directory Project (ODP) [8] is a vast directory of Web pages, edited by over 71,990 volunteer human editors. The listing consists of over five million sites classified into 590,000 categories. With so many different editors from all over the world, there is room for inconsistency in categorizing and a possibility that a site can have duplicate entries. In spite of these inherent characteristics, the ODP directory is one of the largest sources of categorized Web data that is freely available. The directory information provided by this directory is used by some of the leading Web search engines. Google's search engine directory is also powered by the ODP, making it another attractive reason for us to depend on ODP data for our corpus.

The directory is organized as a tree of category nodes. The child nodes are subcategories of the parent. A description for each category node is provided. Each category is associated with a list of editor reviewed sites. The screenshot of the ODP Web page shown in Figure 3.1 shows a glimpse of the structure of the directory.

3.1.2 Categorized Dataset Acquisition

The process we adopted to create the corpus is outlined in Figure 3.2. We describe the process in the following paragraphs.



Arts
Movies, Television, Music...

Business
Jobs, Real Estate, Investing...

Computers
Internet, Software, Hardware...

Games
Video Games, RPGs, Gambling...

Health
Fitness, Medicine, Alternative...

Home
Family, Consumers, Cooking...

Kids and Teens
Arts, School Time, Teen Life...

News
Media, Newspapers, Weather...

Recreation
Travel, Food, Outdoors, Humor...

Reference
Maps, Education, Libraries...

Regional
US, Canada, UK, Europe...

Science
Biology, Psychology, Physics...

Shopping
Autos, Clothing, Gifts...

Society
People, Religion, Issues...

Sports
Baseball, Soccer, Basketball...

World
Deutsch, Español, Français, Italiano, Japanese, Nederlands, Polska, Dansk, Svenska...

Figure 3.1 ODP screenshot (http://dmoz.org/) shows the directory structure.

3.1.2.1 Selection of Top-level Categories

The first step in generating a new categorized data collection is to identify the scope of the category coverage. In our case, since the ODP directory structure defined our scope, we decided to first select the top-level categories from this structure. The categories that we selected and those which we did not are shown in Table 3.1. Our goal in generating this corpus was primarily to generate domain specific datasets. The ODP directory proved to be well-suited to this objective. We excluded the *Top/Adult* (not shown in Figure 3.1) to eliminate any inappropriate pages. We did not consider *Top/Kids & Teens* because it did not express any new concept-based information that was not included elsewhere. We did not select *Top/Regional* as many of the concept topics represented within this vast category were covered in other categories that we selected. Finally, the *Top/World* category lists sites in languages other than English, which was beyond our requirements. Thus, we were left with 13 top-level categories. At the second-level, we performed some

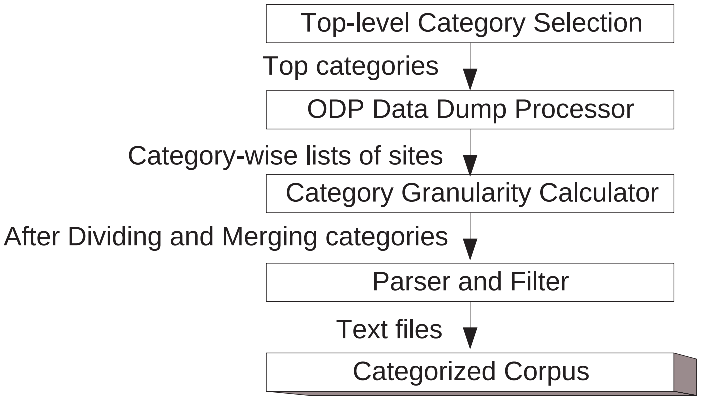

Top-level Category Selection
Top categories
ODP Data Dump Processor
Category-wise lists of sites
Category Granularity Calculator
After Dividing and Merging categories
Parser and Filter
Text files
Categorized Corpus

Figure 3.2 Overview of the corpus generation process.

pruning by disregarding the *Top/News/Online_Archives* category as news stories did not add any new concept topics.

3.1.2.2 Data Dump Processor

The ODP directory data is available as resource description format (RDF) dumps from the ODP Web site. The *ODP data dump processor* module reads the lists of sites by categories and downloads the lists to the local system. We decided to have a minimum depth of at least two levels and a maximum depth of up to three levels. The module downloads any categories at deeper levels (children) as part of the deepest valid category (ancestor). For example, while *Top/Sports/Baseball/Major_League* is a valid independent category, the category *Top/Sports/Baseball/Major_League/Stadiums* has depth of 4 and is too fine grained for our requirements. So the sites belonging to the category *Top/Sports/Baseball/Major_League/Stadiums* are assigned to the category *Top/Sports/Baseball/Major_League*.

The output of this module is a categorized list of sites having a directory structure with maximum category depth of 3, *Top* being at depth 0. For convenience, we replaced all occurrences of *"and"* in category names by *"_"*.

Table 3.1 The Top-level Categories and our Selection

Top ODP Categories	
Selected	**Not Used**
Top/Arts	*Top/Adult*
Top/Business	*Top/Kids & Teens*
Top/Computers	*Top/Regional*
Top/Games	*Top/World*
Top/Health	
Top/Home	
Top/News	
Top/Recreation	
Top/Reference	
Top/Science	
Top/Shopping	
Top/Society	
Top/Sports	

3.1.2.3 Category Granularity Calculator

After having the set of categorized sites, trimmed to our requirements, we finalized the size parameters of the categories. This was closely related with the effective granularity of the corpus. Our approach was to have a final category depth of 2 but after ensuring that the categories were within some size thresholds. The metric used to measure the size of a category was the number of sites listed within that category and all of its children. To help the reader understand the meaning of category depth or levels, we list some example in Table 3.2.

The process consists of two parts: first *dividing* and then *merging*.

During the *dividing* phase, any category with site listing above a threshold of 10,000 sites is progressively split into the child categories until the residual size is above this threshold and the child category being processed has over 2,000 sites. For example, *Top/Science/Social_Sciences* was split into four categories as shown in Table

Table 3.2 The Types of Category Levels with Examples

Level	**Example**
Level 0	*Top*
Level 1	*Top/Science*
Level 2	*Top/Science/Social_Sciences*
Level 3	*Top/Science/Social_Sciences/Archaeology*

3.3. During the *merging* phase, the criterion for allowing a level-2 category to be independent is that it should have over 1,000 sites in it's (including all children) list. If categories do not satisfy this criterion, we merge all such categories (with the same parent) to form a new category with the name of the parent with the suffix “*_Other*” appended. For example, several level 2 categories were merged to form the merged category *Top/Science_Other* as shown in Table 3.4.

Table 3.3 Dividing a Category

Original Category	**Categories after Dividing**
Top/Science/Social_Sciences	*Top/Science/Social_Sciences* *Top/Science/Social_Sciences_Archaeology* *Top/Science/Social_Sciences_Linguistics* *Top/Science/Social_Sciences_Psychology*

3.1.2.4 Parser and Filter

The *parser and filter* module takes sites from the pre-processed directory list and downloads the text of each Web page. We used HTMLParser [38], which is a fast parser for real-time parsing of HTML documents. We limited our downloads only to the page directly pointed by the ODP site listing. We did not download any HTML documents; instead, we downloaded the online, parsed text as a document directly. Before down-

Table 3.4 Merging Categories

Original Categories	**Category after Merging**
Top/Science/Science_in_Society	*Top/Science_Other*
Top/Science/Institutions	
Top/Science/Reference	
Top/Science/Anomalies_Alternative_Science	
Top/Science/Directories	
Top/Science/Software	
Top/Science/Publications	
Top/Science/Educational_Resources	
Top/Science/Employment	
Top/Science/Conferences	
Top/Science/Methods_Techniques	
Top/Science/Chats_Forums	
Top/Science/News_Media	
Top/Science/Search_Engines	

loading any document, we filtered documents not meeting our minimum and maximum parsed size limits of 1 Kbyte and 1 Mbyte respectively. We also limited our corpus to documents that were parseable by the parser, leaving out PDF, PostScript, etc. types. In order to eliminate unwanted documents and to allow only the documents with useful and meaningful text, we applied an innovative method to filter documents by content. We set a minimum *stop word*[1] count of 2 for parsed content of every document to satisfy before downloading it. The resulting downloaded collection was largely free of noise such as error messages, binary files, and unsuitable content.

3.1.3 Corpus Features

The final corpus taxonomy consists of 295 categories with the selected thirteen level-1 categories, each containing a number of level-2 categories obtained after dividing

[1] Stop words are very common words like *and*, *for*, *the*, etc. which do not contain any useful information from an IR perspective.

and merging. The level categories themselves do not contain any corpus documents; instead they only act as parent directories for the level 2 categories. By having a granularity or depth of 2, we try to keep concept topic classification simple. If the classification is too deep, it can divide concept topics accross categories and make the collection too fine grained. Some important comparisons between our UTA-295 ODP corpus and the benchmark corpora (explained in section 3.2) are shown in Table 3.5. Appendix A contains the complete listing of our categories.

Table 3.5 Corpora Comparisons

Corpus	**Contents**
UTA-295 ODP corpus	916,682 Web pages
RCV1	806,791 news stories
BioMed Central's corpus	Over 13,000 research documents

3.2 Overview of Benchmark Collections

We used two benchmark text collections to compare the performance of our concept-based search approach accross corpora. It also gives us an opportunity to evaluate the performance of our corpus with respect to standard collections.

The Reuters corpus volume 1 in English language (RCV1) [26, 39] is a large collection of news stories used for research in IR, NLP, and machine learning. Many studies in text categorization and IR use RCV1 as a benchmark. Documents are assigned type codes from three sets: Topic, Industry, and Region. In our research, we use only the GENT (Arts, Culture, Entertainment), GHEA (Health), GSCI (Science and Technology), GSPO (Sports), GTOUR (Travel and Tourism) categories from RCV1.

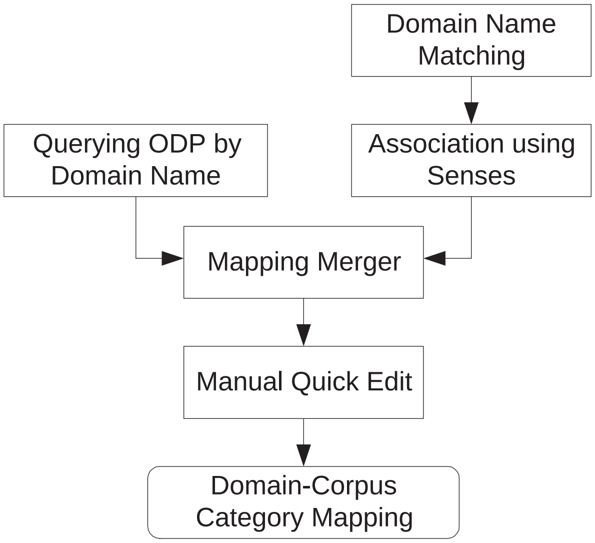


Figure 3.3 Overview of the domain category mapping process.

BioMed Central's research article corpus [27] consists of over 13,000 full-text articles of biomedical research and is widely used for Data Mining. The reason for using this corpus in our system is to evaluate the effect of using a highly subject-specific corpus and comparison of results with UTA-295 ODP corpus for this subject area.

3.3 Domain-category Mapping

In order to evaluate the effects of supplementing concept representation and query expansion by the use of lexical resources, we need to have some kind of a relation between the lexical components and the text corpora. A mapping between semantic domains and the corpus categories could serve this purpose, there by enabling us to combine the human knowledge embedded in the lexical resources with the statistical analysis of text corpus. In this section, we describe the methodology to create a mapping between the lexical, semantic domains and our Web corpus categories.

The process is depicted in Figure 3.3. Mappings obtained by multiple ways are combined to arrive at the final mapping. We did not map the domains *factotum*, *color*, *quality*, and *number* due to their generality.

3.3.1 Identifying the Mappings

First, the ODP search engine was queried for every domain in the WordNet Domains 3.1 database. The results of ODP search were manually rated as relevant or irrelevant and a set of relevant categories C_1 for the domain was stored. In all, 164 domains were mapped to a total of 434 UTA-295 ODP corpus categories (including repetitions).

In the direct *domain named mapping* phase, each domain of WordNet Domains 3.1 is compared with the name of every category from our UTA-295 ODP corpus. If the domain taken as a string matches any token of a category name, a match is recorded and stored in the list C_{21}.

The *sense association* phase is more intensive and is based on the automatic association of Web directories with word senses [40]. We put forth the logic in Algorithm 1, which uses functions from Algorithm 2 and Algorithm 3. For every domain d, we first get the domain name matched categories and store it as C_{21}. In order to perform domain category association based on word senses, we use up to 100 representative nouns from every domain. The reason for using nouns is the observation that most ambiguous query terms in real world searches are nouns. For every word, we construct a query q consisting of the word, a synonym, and a hypernym as shown in Algorithm 2. This query is used to search the ODP search engine, which is represented by the function GET-CATEGORIES-BY-ODP-SEARCH in the algorithm.

The related terms L of the word including synonyms, hypernyms, hyponyms, holonyms, meronyms, and coordinate terms are obtained from the WordNet database using JWNL [41]. For every category obtained from the ODP search, we compute a

Algorithm 1 Mapping WordNet Domain to ODP Categories

```
**Input:** Domain $d$ with any categories $C_{21}$ mapped using domain name matching.
**Output:** A List of ODP categories $C_2$ for the domain $d$.
define new lists $domainT$ and $domainS$ to store categories and scores for domain.
**if** $length[C_{21}] \ < \ 3$ **then**
    $W \ \Leftarrow$ up to first 100 sense tagged nouns from domain $d$
    **for** $i \ \Leftarrow \ 0$ to $length[W] \ - \ 1$ **do**
        ▷ GET-CATEGORIES-BY-ODP-SEARCH is a call to ODP search engine.
        $T \ \Leftarrow$ GET-CATEGORIES-BY-ODP-SEARCH(GET-QUERY($W[i]$))
        ▷ GET-LEXICAL-TERMS is a call to WordNet to obtain synonyms, hyponyms,
        hypernyms, holonyms, meronyms, and coordinate terms for $w$.
        $L \ \Leftarrow$ GET-LEXICAL-TERMS($w$)
        $S \ \Leftarrow$ GET-SCORES($w$, $L$, $d$, $T$)
        **for** $j \ \Leftarrow \ 0$ to $length[T] \ - \ 1$ **do**
            $index \ \Leftarrow \ indexOf[domainT, \ T[i]]$
            **if** $index \ \neq \ -1$ **then**
                $domainS[index] \ \Leftarrow \ domainS[index] \ + \ S[j]$
            **else**
                $add[domainT, \ T[j]]$
                $add[domainS, \ S[j]]$
$maxScore \ \Leftarrow \ (sort[domainS])[length[domainS] \ - \ 1]$
$_{22} \ \Leftarrow$ all $domainT[j]$ such that $domainS[j] \ = \ maxScore$
$C_2 \ \Leftarrow \ C_{21} \ \cup \ _{22}$
return $C_2$    ▷ return the categories for this domain
```

score by finding word-word co-occurrences between each of the related terms and category parts (obtained after splitting the category at every "_"). For each domain, the score is computed by adding the individual scores for words. At the end of the process, for every domain, the categories C_{22} with the maximum score $maxScore$ are merged with C_{21} using the union operation to get the final domain category mapping.

In all, 164 domains were mapped to a total of 367 categories.

3.3.2 Merging and Quick Editing

For each domain, the categories obtained from the domain querying, C_1, and those obtained from the combination of domain name matching and sense association, C_2 merged by simply taking the union to get the merged mapping.

$$C \Leftarrow C_1 \cup C_2$$

Although the automatic mapping helps us broaden our coverage beyond the obvious mappings, it also results in some highly unwanted mappings. These mappings, however small, can cause a significant deviation from the semantic meaning of a subject or concept topic. Hence, we perform a final *quick edit*, which is a simple fast editing to delete any of these mappings. The final mappings consisted of 164 domains mapped to 697 categories and are shown in Appendix B.

3.4 Summary

In this chapter, we first described our approach to construct a Web corpus consisting of 295 categories using the ODP Web directories. We refer to this categorized text collection as UTA-295 ODP corpus. We explained the process of identifying the directory categories and fitting all the categories within a pre-determined size by dividing or merging the standard categories. We told about the filtering and downloading of

Algorithm 2 GET-QUERY used in mapping WordNet Domain to ODP Categories

```
**Input:** Sense tagged Word (noun) $w$.
**Output:** String $q$ to be used for searching ODP search engine.
▷ GET-SYNONYMS is a call to WordNet using JWNL.
$S \Leftarrow$ GET-SYNONYMS($w$)
**for** $i \Leftarrow 0$ to $length[S] - 1$ **do**
  **if** $lemma[w] \neq lemma[S[i]]$ **then**
    $syn \Leftarrow lemma[S[i]]$
    **break**
▷ GET-HYPERNYMS is a call to WordNet.
$H \Leftarrow$ GET-HYPERNYMS($w$)
**for** $i \Leftarrow 0$ to $length[H] - 1$ **do**
  **if** $lemma[w] \neq lemma[H[i]]$ **then**
    $hype \Leftarrow lemma[H[i]]$
    **break**
$q \Leftarrow lemma[w]$
**if** $syn \neq$ EMPTY and $hype \neq$ EMPTY **then**
  $q \Leftarrow q +$ “ ” $+ lemma[syn] +$ “ or ” $lemma[hype]$
**else if** $syn \neq$ EMPTY **then**
  $q \Leftarrow q +$ “ ” $+ lemma[syn]$
**else if** $hype \neq$ EMPTY **then**
  $q \Leftarrow q +$ “ ” $+ lemma[hype]$
return $q$
```

Algorithm 3 GET-SCORES used in mapping WordNet Domain to ODP Categories

```
Input: The word $w$, lexical terms $L$, domain $d$, and categories $T$.
Output: List $S$ of scores (corresponding to categories of $T$).
for $i \Leftarrow 0$ to $length[T] - 1$ do
    $P \Leftarrow split[T[i]]$ using "_" into parts.
    $S[i] \Leftarrow$ number of co-occurrences between elements from $P$ and $L$.
return $S$
```

the documents from the Web and the tools we used. The benchmark corpora that we used are listed along with brief information about their features. We also delved into the details of the method used to generate a mapping between the WordNet Domains and the categories within our corpus. A multi-step process which includes multiple ways of domain-category association generation followed by result merging and a quick verification editing was presented. It was accompanied by the formal presentation of the algorithms used.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

Concept-based search is a module which builds over an existing text search engine and provides enhanced concept-based functionality. Internally, it identifies the possible subject areas from the user query and expands the query into these areas. The scope of concept topics (subject areas) identification is limited to the corpora that are pre-trained and the lexical resources that are used. Use of lexical resources enables the module to produce better word sense discrimination and domain identification. When a domain-specific search is executed by a user, the query is expanded using the related terms for the terms in the query. The related terms are obtained from the word vector space using a variant of latent semantic indexing approach. The results of the expanded query are obtained using the underlying search engine. In this chapter, we explain the overall architecture and the implementation details of the system.

4.1 Concept-based Search Steps

Concept-based search is a two step process. We first explain from the user point of view and then the system-side perspective.

4.1.1 User Perspective

The concept-based search process as a user session is shown in Figure 4.1. The user first enters a search query. The system returns possible concept topics which are subject areas, along with the results for a simple search. The user selects one of the

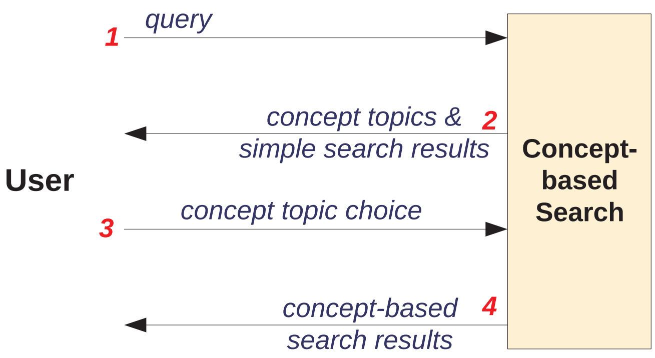


Figure 4.1 User perspective: concept-based search steps.

concept topics and the query with the concept topic is re-submitted. Concept-based search is performed and the results are returned to the user.

4.1.2 System Perspective

We explain the two parts or steps at the system-side briefly to enable the reader to get a working understanding of the system.

STEP 1: In the *concept topics identification* step, all the relevant concept topics for the user query are identified and the user is presented with these concept topics along with the search results of a simple Web search for the query. For a given query, this step is performed only once and is followed by search within these concept topics, which are essentially subject areas.

STEP 2: This is the *query expansion* step but can also be called concept-based search step. The user request consists of the query and one concept topic as the area of the search. The query is expanded in this concept space and the Web search results for this transformed query are returned. If another concept topic is selected, this step is repeated for the new concept topic.

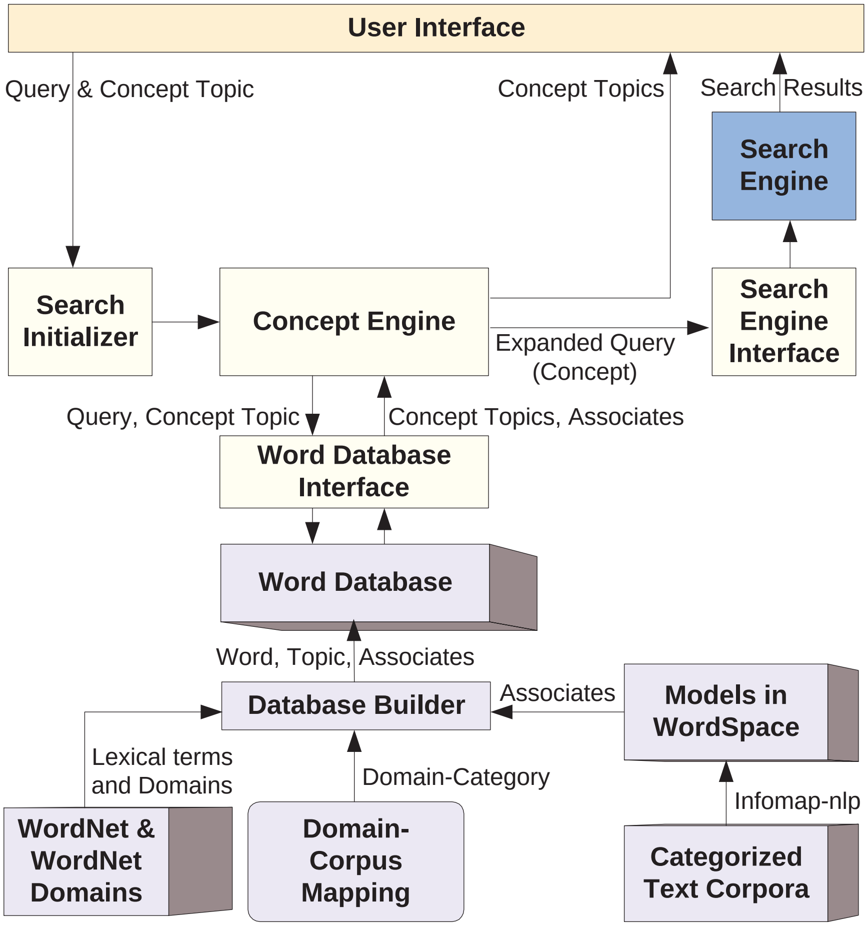


Figure 4.2 Concept-based search architecture.

It is important to note that although the concept topic selection is interactive, the query expansion itself is automatic.

4.2 Architecture and Implementation Details

The system consists of two main sections: the *concept topics identification* and *query expansion* functions together as one section, which is the primary contribution of this thesis, and the *search engine section*, which is any search engine which provides an interface for search service. The architecture is shown in Figure 4.2. The user submits a search query consisting of English words. The *search initializer* validates the input query and keeps track of what step for the current query is being executed.

If it is the initial *concept topics identification* step, then the *concept engine* module looks up the *word database* using the *database interface* and retrieves the concept topics to which the query belongs. It returns these concept topics and the search results from the *search engine* for the original query to the user.

If it is the subsequent *query expansion* step, then the *concept engine* module looks up the *word database* using the *word interface* and retrieves the associated words for the query. The *search engine* is invoked for the expanded query and the results are displayed to the user. The user can also search for the query in another concept topic and this step is repeated.

4.2.1 Search Initializer

The *search initializer* module first checks the user query for valid characters. If there are invalid characters, the search process is aborted with an appropriate message to the user. A simple validator serves the purpose here as the underlying search engine is assumed to have a validator as per its requirements. This module also saves the state of query processing of the current query request. In other words, it keeps track of the state of this process and passes this information to the *concept engine*.

In the implementation, the *search initializer* is embedded within the Web application to provide for basic query validation. It also keeps track of the state of the search by using the user's session information.

4.2.2 Concept Engine

This module is the main controller for the entire search process. It co-ordinates with the *search initializer*, the *word database interface*, and the *search engine interface*. It performs different jobs in each of the two steps.

During the initial *concept topics identification*, this module does not perform any query expansion. It tokenizes the query into individual words and first retrieves the concept topics associated with each token. This enables all the concept topics to be available and parallel query expansion can be performed in the next step for the same query. For this, it utilizes the services of the *word database interface* module. The objective here is to obtain the set of concept topics for the query as a whole. If the query is a single token, that is, a word, then the concept topics for that word are obtained from the word database. The method GET-CONCEPT-TOPICS of the *word database interface* is invoked. If the query consists of greater than one token, for each token, the concept topics are retrieved and each concept topic in the combined set is assigned a score depending on the number of tokens that belong to that concept topic. The concept topics with the maximum score are selected. The algorithm is shown in Algorithm 4. As we assume that our approach is limited to processing short queries, we assume that the number of query terms is less than 4. Also, we limit the maximum number of concept topics per word to 7.

During the *query expansion* stage, the inputs to the concept engine are the concept topics for the query and the query itself. For each token, the *concept engine* retrieves the associate terms from the *word database* using the call GET-CONCEPT-ASSOCIATES on the *word database interface*. For the simple case of one word query, the expanded query consists of the input word and the associates up to the maximum query length or *maxExpQueryLength*. The expanded terms are readily available for all the parallel concept topics identified for the query. The value of *maxExpQueryLength* is decided

Algorithm 4 Concept Topics Identification for Input Query

```
Input: User query $query$ (consisting of one or more tokens).
Output: A List of concept topics (domains or categories) for the query.
$Q \Leftarrow tokenize(query)$
if $length[Q] = 1$ then
    return GET-CONCEPT-TOPICS($query$)    ▷ retrieves from word database
else
    $max \Leftarrow 0$
    for $i \Leftarrow 0$ to $length[Q] - 1$ do
        $T \Leftarrow$ GET-CONCEPT-TOPICS($Q[i]$)    ▷ concept topics for this token
        for each $conceptTopic \in T$ do
            if $containsKey(H, conceptTopic) =$ TRUE then
                $count \Leftarrow value[(get(H, conceptTopic)]$
                $count \Leftarrow count + 1$
                $update(H, conceptTopic, count)$
                if $count > max$ then
                    $max \Leftarrow count$
            else
                $insert(H, conceptTopic, 1)$
                if $max = 0$ then
                    $max \Leftarrow 1$
        $i \Leftarrow 0$
        for each $x \in H$ do
            if $max = value[x]$ then
                $C[i] \Leftarrow key[x]$
                $i \Leftarrow i + 1$
    return $C$
```

empirically but is also limited by the length supported by the underlying search engine. When the query consists of more than one words (or tokens), the combined expansion terms representing each word of the input query are computed subject to the overall limit *maxExpQueryLength*. This is further explained in Algorithm 5. The final query is a Boolean combination of the original user query and the expansion terms, using the AND and OR operators as shown in the Figure 4.3. Like most other implementations in this

Algorithm 5 Obtaining Query Expansion Terms for Input Query and Concept Topic

```
Input: User query $query$ (consisting of one or more tokens) and concept topic $conceptTopic$.
Output: A List of words for query expansion.
$Q \Leftarrow tokenize(query)$
if $length[Q] = 1$ then
    return GET-CONCEPT-ASSOCIATES($query$)    ▷ retrieves from word database
else
    $expandedQuery \Leftarrow query +$ " ("
    $tokenDelimiter \Leftarrow$ " OR "    ▷ to make the query less stringent
    $maxExpQueryLength \Leftarrow 9$    ▷ 9 for our experiments
    $maxAssocPerToken \Leftarrow (maxExpQueryLength - length[Q])/length[Q]$
    for $i \Leftarrow 0$ to $length[Q] - 1$ do
        $T[i] \Leftarrow$ GET-CONCEPT-ASSOCIATES($Q[i]$)    ▷ $T$ is the two dimensional array containing associates for every token of $query$
        for $j \Leftarrow 0$ to min($length[T] - 1, maxAssocPerToken$) do
            $expandedQuery \Leftarrow expandedQuery + tokenDelimiter + T[i][j]$
    return $expandedQuery +$ ")"
```

thesis, the *concept engine* is implemented in Java 2 Standard Edition 5.0. The algorithms explained above are coded for the concept domains and categories (for UTA-295 ODP Web corpus and the benchmark corpora). For each of these, a different type of request is invoked in the *database interface* module but the functionality remains the same.

4.2.3 Word Database Interface

The *word database interface* provides the functionality to retrieve information from the database. The available functionalities include reading concept topics for a word: GET-CONCEPT-TOPICS, and reading associates for a word-topic pair: GET-CONCEPT-ASSOCIATES. The interface abstracts the database system from the run-time system and provides for simple invocations to serve the *concept engine*. The database environment, it's structure, etc. can be modified by making changes in the interface implementation. Other modules remain unaffected.

On the implementation side, the *word database interface* is a Java class called `WordDatabaseInterface`, which is primarily responsible for invoking any database methods. The value of the attribute `wordDbEnvPath` is read from a properties file and can be modified to use any desired database.

4.2.4 Word Database

We present the conceptual details and the implementation details separately.

4.2.4.1 Word Database Conceptual Details

The *word database* is essentially an index where the words are the *keys* and the concept topics and the corresponding related words (associates) constitute the *data*. The database is indexed using words from the text corpora used in the system. Once created, the operations performed on the database by the database interface are look-up operations. In the pre-processing stage, which is carried out offline, the database is populated with the words and their associates. The *database builder* creates all the word entries and updates them with related information.

Each entry or *tuple* in the word database consists of the following attributes:
<*word, all_concept_models, all_normalized_term_frequencies,*
concept_category_1,..., concept_category_n,
associates_category_1,..., associates_category_n,
concept_domain_1,..., concept_domain_n,
associates_domain_1,..., associates_domain_n,
wordNetDomainFlag>

After the database is populated, the attributes *all_concept_models* and *all_normalized_term_frequencies* are not used.

4.2.4.2 Word Database Implementation Details

We use Berkeley DB Java Edition 2.1.30, a pure Java, transactional data storage engine. The important features that are of particular benefit to our approach include in-process execution, application native storage, internal storage using B-tree providing quick access, and efficient caching for the most frequently used data.

The basic format of a database record is defined and encapsulated by an instance of the class `WordData`. Other classes required for the environment and binding are also implemented.

4.2.5 Database Builder

The database is created using inputs from multiple sources of information. The *database builder* module is responsible for the creation and population of the database.

It runs offline and prepares the database for use in the system. Once ready the database does not need any updating services from this module.

The *database builder* operates in two steps, viz., *create words* and *update words*. These steps are described below.

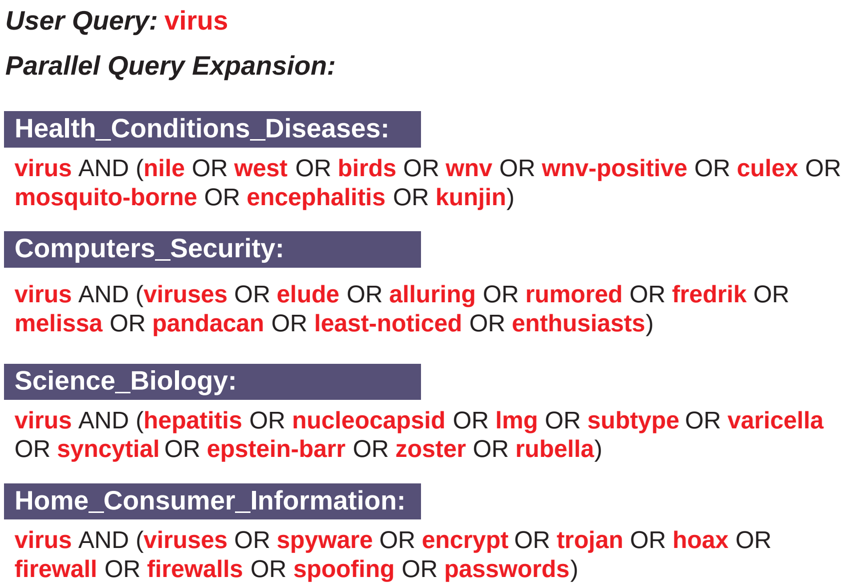
4.2.5.1 Create Words

During the creation step, words from the text corpora models are added to the *word database*. The words from the dictionary of the model corresponding to each corpus category are added one-by-one. Basic character-based filtering removes words with unwanted characters. The models to which a word belongs and the corresponding *normalized term frequencies* are stored in the two attributes *all_concept_models* and *all_normalized_term_frequencies* respectively. For every word, the *normalized term frequency in a model* of documents is given by

$$normalized\ term\ frequency\ in\ model = \frac{term\ frequency\ in\ model}{maximum\ term\ frequency\ in\ model}$$

where $term\ frequency\ in\ model$ is the sum of frequencies of occurrence of the term (word) within all the documents of the model and the $maximum\ term\ frequency\ in\ model$ is the maximum of all term frequencies, excluding stop words. It is worth noting here that a model consists of a collection of documents.

The program for this functionality is executed once to create the database after all the models have been already created. The program reads the *dictionary* file of every model created (explained in section 4.2.6.4) and adds words to the database. If a word exists the data for that word is updated to reflect it's existence in the current model, along with the *normalized term frequency* value. While reading the dictionary, we exclude stop words and any word that occurs in fewer than 3 documents in that model.



User Query: **virus**

Parallel Query Expansion:

Health_Conditions_Diseases:

virus AND (**nile** OR **west** OR **birds** OR **wnv** OR **wnv-positive** OR **culex** OR **mosquito-borne** OR **encephalitis** OR **kunjin**)

Computers_Security:

virus AND (**viruses** OR **elude** OR **alluring** OR **rumored** OR **fredrik** OR **melissa** OR **pandacan** OR **least-noticed** OR **enthusiasts**)

Science_Biology:

virus AND (**hepatitis** OR **nucleocapsid** OR **lmg** OR **subtype** OR **varicella** OR **syncytial** OR **epstein-barr** OR **zoster** OR **rubella**)

Home_Consumer_Information:

virus AND (**viruses** OR **spyware** OR **encrypt** OR **trojan** OR **hoax** OR **firewall** OR **firewalls** OR **spoofing** OR **passwords**)

Figure 4.3 Parallel query expansion using associates for the word *virus*

4.2.5.2 Update Words

During this phase, each word entry in the database is re-visited and updates for the word data are made. For every concept category, the model with the highest *normalized term frequency* is selected as the representative concept model. These concept topic names are saved in *concept_category_1,..., concept_category_n*. Corresponding associates for the word are obtained from the model using the Infomap-nlp software [17] and saved in *associates_category_1,..., associates_category_n*. Figure 4.3 shows an example of parallel query expansion using associates for the word *virus* which is a prospective user query.

In order to set concept domains, the synset for the word is first obtained from WordNet. WordNet Domains are used to look-up the domains for the synsets. In case of duplicate domains, only the first one is considered. The models mapped to a concept domain are retrieved from the *domain corpus mapping*. These domain names are saved in *concept_domain_1,..., concept_domain_n*. For every concept domain, the mapped model with the highest *normalized term frequency* is selected as the representative concept model. The associates for the word in the selected model are obtained from the model using the Infomap-nlp software [17]. In addition, the first hypernym for the word is also appended to the associates to take maximum advantage of the lexical resource and also to enable us to observe whether this has any significant positive effects on the results. The terms are stored in *associates_domain_1,..., associates_domain_n*.

The program for the *update* step invokes the JWNL [41] API to interface with the WordNet dictionary. It also reads the WordNet Domains database, which is in the form of a flat file. The most important task performed in this step is invoking the `associate` command of the Infomap-nlp software and get the associates for every word. As we will see in the remaining part of this chapter, the models trained by Infomap-nlp contain word vectors and their similarities can be measured using the `associate` command. For every word, we take the *associates* having similarity value above a threshold of 0.5. The similarity value is the cosine of the angle between the vectors in WordSpace for the two words. When this process ends, we are left with a fully populated database, ready to be used for our online experiment.

It is worth mentioning that we indexed on words, not phrases. Also, we eliminated stop words and words with hyphens or quotes. Table 4.1 gives some statistical information about the words in the *word database* and the comparison becomes very clear from Figure 4.4. The total number of words is not the sum of the individual contributions from the source corpora because there are words that are common to two or more corpora.

Table 4.1 Word Database Statistics

Source	**Words in the Database**
UTA-295 ODP Web corpus	310,037
RCV1 (selected categories)	42,952
BioMed Central corpus	28,923
Total words	**320,843**

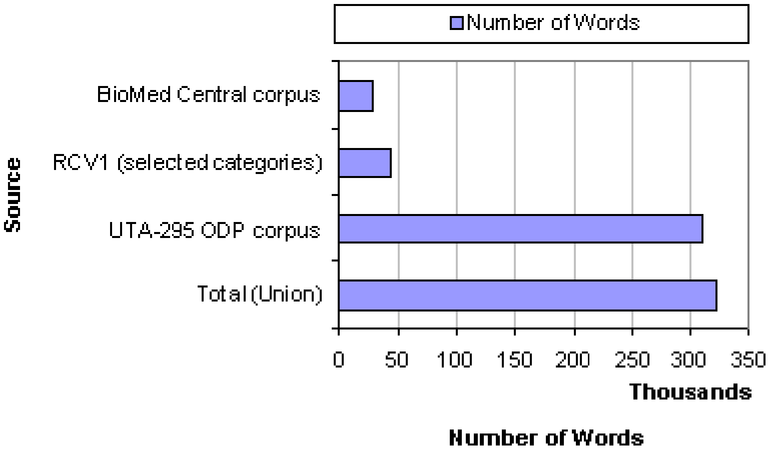


Figure 4.4 Chart showing the word contributions from various corpora.

4.2.6 Database Resources

The *database builder* uses various resources to build the *word database*. We briefly discuss these below. More details of sections 4.2.6.1 and 4.2.6.3 are explained in Chapter 3.

4.2.6.1 Domain Corpus Mapping

The domain corpus mapping gives the mapping from a WordNet Domain domain to one or more corpus models, which are one of the categories obtained from the Open Directory Project [8]. An entry in this mapping file is of the form: "domain_name: set of models". For example, the *athletics* domain is mapped to *Sports_Running* and *Sports_Track_Field*. The entry is as follows:

athletics: Sports_Running Sports_Track_Field

4.2.6.2 Lexical Resources

We use WordNet [13] and WordNet Domains [14] as our lexical references to enhance our search process. An overview of these resources is given in Chapter 2. We use WordNet to find the senses of a word. The word senses have to be transformed to the corresponding domains, which is one way to represent the concept topics. This is achieved by using the word sense-to-domain mapping provided by WordNet Domains.

4.2.6.3 Domain-specific Text Corpora

The system uses domain-specific text corpora from the ODP, viz., UTA-295 ODP corpus, and also two benchmark text corpora, viz., Reuters and BioMed Central to enable comparison between the two corpora types. The corpora should be plain text representations with granularity to enable better distinctions within a particular domain.

4.2.6.4 Models in WordSpace

The text corpora are used to train the models in WordSpace [16]. The number of models is equal to the number of text corpora that we have, which in turn depends on the extent to which we can obtain granular corpora. The models are trained in the pre-processing step, which is done offline. Once trained, we use the models to find related words for a given word during the *Update Words* step explained in 4.2.5.2.

4.2.7 Search Engine Interface and Search Engine

The *search engine interface* provides an abstraction for accessing the actual *search engine*. In particular, it provides the *concept engine* the functionality to call the search methods of the *search engine*. The query is passed as input and the search results are returned. Only this interface may have to be changed to use another search engine, thus making the system adaptable to practically any search engine.

We use the Google search engine through the Google Web API service. A class called `SearchEngine` represents the interface with the Web API service. It reads the access key from a properties file.

4.2.8 User Interface

The *user interface* is an important part of the system because it represents a part of the service offering to the user. In Web search, the user interface design should ensure that the user is not required to make too many selections. In other words, there should be a balance between interactiveness and automation.

During the *concept topics identification* phase, the user is provided with a simple interface that has an input text box and a submit button. During the *concept-based search* phase, the user is presented with the identified concept topics and is expected to click on one of these.

The *user interface* is a Web-based client just like a search engine's interface. In the *concept topics identification* phase of the search, the interface provides only a text box and a submit button. In the *query expansion* phase, the interface presents various concept topics for the user to select.

A simple implementation using JavaServer Pages (JSP) was used for the user interface. The Web application ran on an Apache Tomcat 5.5.15 servlet container on a dedicated computer system.

4.3 Summary

This chapter gave a description along with a diagram of the architecture of the concept-based search system. The functionality of different modules, viz., the *search* *initializer*, the *concept engine*, the *word database interface*, the *database builder*, the *search engine interface*, and the *user interface* was described. The roles of the *word database*, the *domain corpus mapping*, the *lexical resources*, the *corpus models*, and the *search engine* were explained. Algorithms used by the concept engine for identifying the concept topics for a search query and for obtaining the query expansion terms were presented. For every functionality, the corresponding implementation details were listed.

CHAPTER 5

EVALUATION AND RESULTS

In the previous chapter, we saw the architecture and the functional details of all the parts of the system. The idea behind the design of this approach was to enable users to obtain relevant search results without having to navigate through irrelevant results. This facility was at the expense of the user being able to choose one of the concept topics presented by the search system. In this chapter, we attempt to explore the implications of this approach in improving user experience. We start by identifying some queries from available references as our standard test queries. We provide subjects with these queries and an interactive system to help them accomplish this task. The relevance feedback of the subjects is saved and analyzed. Finally, we present and discuss the results.

5.1 Evaluation

5.1.1 Evaluation Tasks

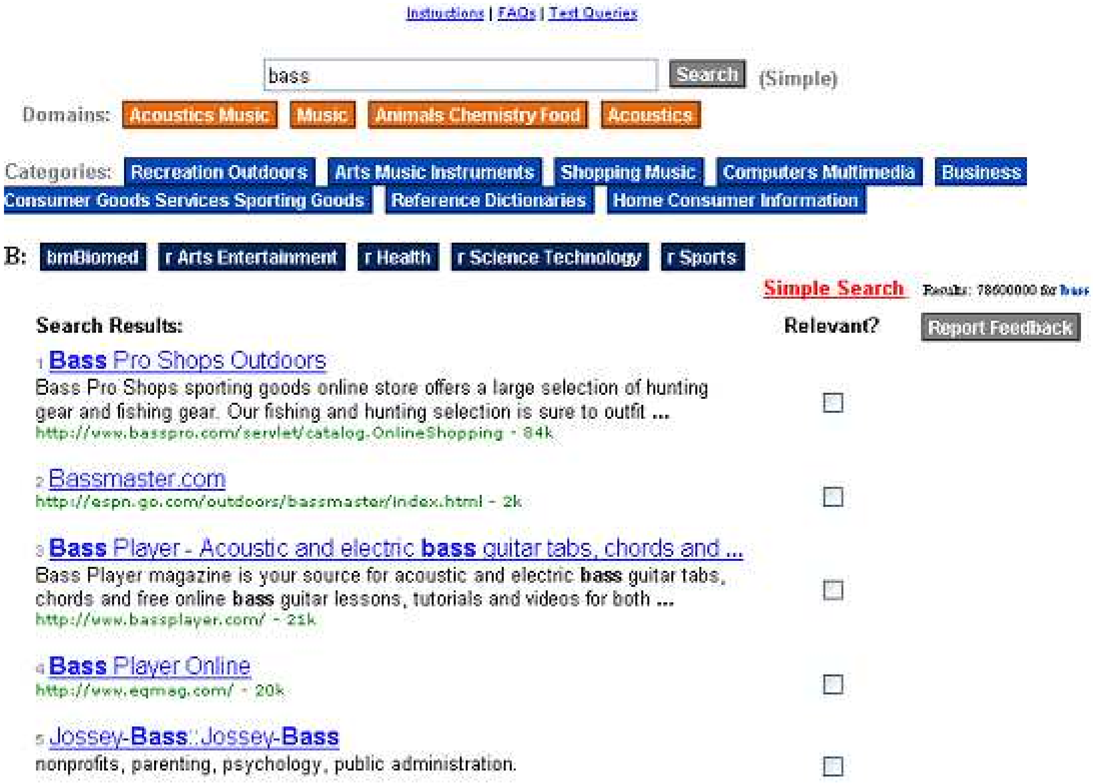
In this thesis, we aimed to tackle the challenge of retrieving most relevant information for the user. To evaluate our approach for this goal, we had to compare it with a baseline approach. We chose Google search as our baseline and decided to compare the results from our approach with those from this commercial search engine. Although we do not present the comparative precision values for search using Ask.com [10], it is important to mention here that the approach used by Ask is interactive query expansion and it outperformed our approach for the few standard queries that we used on it.

In addition to this comparison, we also wanted to compare the results of using different corpora and resources within our approach. To this end, one task was to compare the concept-based search results of query expansion with and without using WordNet and WordNet Domains. Another task was to compare the results for query expansion using our UTA-295 ODP corpus and the benchmark corpora [27, 42, 39]. Finally, to evaluate the time required for the users to retrieve relevant results we measured the time for identifying first 10 relevant results.

We prepared a set of one word queries by including English words from the Senseval 3 competition [43], the TWA Sensed Tagged Data [44, 45], and [46]. These were used as standard test queries. The concept domains were the domains from WordNet Domains while the concept categories were directly derived from the corpus models from the UTA-295 ODP corpus, RCV1 [42, 39], and BioMed Central's corpus [27]. The list of these 88 standard queries is shown in Table C.1 of Appendix C. In all, there were 164 concept domains, 295 concept categories from the UTA-295 ODP corpus, five concept categories from the RCV1, and one concept category for the BioMed Central's corpus.

5.1.2 Evaluators

We located subjects by distributing flyers on the university campus. There were 18 respondents who were given printed and electronic instructions to understand the tasks and optionally some were trained with a demonstrative example. The users were told to evaluate only as much as they could and were free to exit at any point. They were also allowed to perform the tasks over multiple sessions. The users were provided with the standard test queries which were all of one word in length. They were also told to enter any other queries that they would in a search engine so that we could test the system with adhoc queries. This was possible because our UTA-295 ODP corpus was broad enough to cover most of the real world queries.



Search Results:

1 **Bass** Pro Shops Outdoors □
Bass Pro Shops sporting goods online store offers a large selection of hunting gear and fishing gear. Our fishing and hunting selection is sure to outfit ...
http://www.basspro.com/servlet/catalog.OnlineShopping - 84k

2 Bassmaster.com □
http://espn.go.com/outdoors/bassmaster/index.html - 2k

3 **Bass** Player - Acoustic and electric **bass** guitar tabs, chords and ... □
Bass Player magazine is your source for acoustic and electric **bass** guitar tabs, chords and free online **bass** guitar lessons, tutorials and videos for both ...
http://www.bassplayer.com/ - 21k

4 **Bass** Player Online □
http://www.eqmag.com/ - 20k

5 Jossey-**Bass**::Jossey-**Bass** □
nonprofits, parenting, psychology, public administration.

Figure 5.1 Screenshot of the user interface.

5.1.3 User Interface and Data Collected

The user interface was dynamic, Web-based powered by JavaServer Pages running on Apache Tomcat. The interface also provided a link to instructions for the user and a link to the list of standard queries. The interfaces for testing benchmark collections had extra buttons corresponding to these collections. A screen shot of the user interface is shown in Figure 5.1. The users were not shown any indication of the query expansion process which was being carried out by the system. The query was entered in a text box and the users clicked the submit button. As the new page loaded, the different concept topics for the query were shown along with up to top 10 results from simple search using the Google search engine. The user could take up a sense of the query as the intended sense and then mark the results as relevant by checking the check boxes. On clicking “Report Feedback”, the user’s response was saved in a database on the server. The users then clicked on the concept topic to get concept-based results and saved their responses in a similar manner. When comparing the results of different methods and types of corpora used by the system, the users were not asked to rate the results for the simple Google search.

The data recorded consisted of the query, the concept topic type the user selected, the number of relevant results, and the total number of results displayed. The test queries (other than the standard queries) entered by the users are listed in Appendix D.

5.2 Results and Analysis

The metric we used was precision for the top results up to a maximum of 10 results (P@10). We decided a cut-off value of 10 because the top results are the most important ones and also to make it easier for users to evaluate more queries in the same amount of time. For evaluating the effectiveness of Web search, recall is not such a good measure because it is difficult to know the total number of relevant results. Here we show the results for each type of experiment.

5.2.1 Concept-based Search Versus Simple Google Search

We asked users to rate the relevance of results from (1) simple Google search, (2) concept-based search using RCV1, and (3) concept-based search using UTA-295 ODP Web corpus. The total number of query instances tried was with 14 unique queries. The results are shown in Table 5.1. We observe that the results for either type of concept-based search have outperformed the simple Google search. This validates our assumption that concept-specific query expansion is significantly superior and effective to standard search. Moreover, the results from our UTA-295 ODP Web corpus have slightly higher

Table 5.1 Comparing Precision for Simple and Concept-based Search Approaches

	Simple Google	**Concept-based**	
		RCV1	**UTA-295 ODP**
P@10	0.297	0.514	0.772

precision than that for results from RCV1. This shows that the quality of our corpus for the purpose of topic-wise query expansion is higher than a standard corpus.

5.2.2 Concept-based Search with and without using WordNet and WordNet Domains

We divided concepts into three types: (1) concept-based search using WordNet and WordNet Domains and UTA-295 ODP Web corpus, (2) concept-based search using UTA-295 ODP Web corpus but without the use of any lexical resources, and (3) concept-based search using benchmark corpora but without the use of any lexical resources. The results are shown in Table 5.2. The difference in the query instances for the three types is because the number of concept topics for each type is not exactly equal and hence the user choices also vary. Also, some users were not required to rate the benchmark categories. Nevertheless we get valuable insight into the comparative performance between the three

Table 5.2 Results for Concept-based Search with and without using WordNet (WN) and WN Domains (WND)

	WN and WND with UTA-295 ODP corpus			**UTA-295 ODP corpus categories**			**Benchmark corpora categories**		
Query Length	1	2	3	1	2	3	1	2	3
Queries	101	22	2	135	15	4	17	13	3
P@10	0.633	0.609	0.000	0.730	0.540	0.575	0.688	0.908	0.933

approaches. For single word queries, the results for each type of concept-based search were above a P@10 value of 0.5. The concept-based approach using only UTA-295 ODP corpus performed slightly better than the other approaches. For a query length of 2, the approach using lexical resources was more effective than the UTA-295 ODP categories approach. Moreover, interestingly, the benchmark categories were highly effective for queries with length of 2. On analyzing the queries, we found that these queries were from the Bio-Medical domain. The results for a query length of 3 are inconclusive due to the lack of number of queries.

5.2.3 Time Taken by Users to Identify First 10 Relevant Results

We measured the time taken by users to identify the first 10 relevant results when they were shown up to 30 results from simple Google search and concept-based search with and without the use of lexical resources. As before, the users were free to use the standard queries or their own search queries. We calculate the average time spent per test query as follows:

$$average\ time\ spent\ per\ query = \frac{total\ time\ spent}{number\ of\ queries}$$

where *total time spent* is the time spent to identify the 10 relevant results for all queries by the users, and $number\ of\ queries$ is the number of queries evaluated by the users. The results are shown in Figure 5.2. From the chart, it is clear that the times required using concept-based approaches are less than that required for simple search. This implies that relevant results are ranked higher due to the concept-based approach. It is interesting to note that the users completed that task for concept-based search without the use of WordNet and WordNet Domains faster than that using WordNet and WordNet Domains. A possible reason for this could be the wider coverage and more granular categorization of the corpus categories.

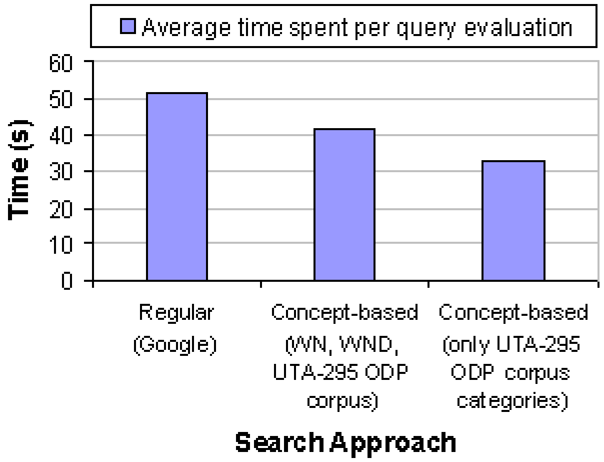


Figure 5.2 Comparing average time spent per query by users.

5.3 Summary

In this chapter, we first introduced the tasks we intended to undertake in the evaluation process. We then gave a description of the users, the user interface, and the evaluation method used by the users. We gave details of the feedback data that we captured from user responses. Finally, we presented the results of the different experiments that we performed along with the analysis for each of them.

CHAPTER 6

CONCLUSIONS

The importance of the Web as a source for information has never been more than what it is today. One can only imagine that it will grow in the future. Web search engines are both the means and the facilitators to access this vast information in an efficient and user-friendly manner. In this work, we addressed the problem of irrelevant results for short queries on Web search engines. Studies on search engine logs indicate that for popular Web queries, the average query length is 1.6 words. Such short queries fail to provide sufficient context to disambiguate possible meanings associated with the search terms. The result is a set of irrelevant Web pages that the user has to filter through navigation and sometimes after examination. This in turn results in lost time and productivity.

For this problem, we proposed and implemented a solution with the following parts: First, we predict the potential domains for the search terms. This prediction is based on word occurrences and relationships observed in the various domains (categories) of a corpus. Next, we expand the search terms in each of the predicted domains in parallel. We then submit separate queries, specialized for each domain, to a general purpose search engine such as Google. The user is presented with categorized search results under the predicted domains.

The theoretical foundations of our approach include concept identification in the form of associated terms through latent semantic indexing, in particular the WordSpace model, one sense per collocation, and one domain per discourse assumptions, and sense disambiguation through sufficient context.

Evaluations based on user judgments indicate that the approach helps users avoid having to examine irrelevant Web search results, especially with shorter queries. This result is quantified by both the average precision for top ten search results as well measurements of time spent until the user is satisfied with search results.

We present the insights gained through a comparative analysis of the use of different corpora for the purpose of domain-specific query expansion. From the user studies, we also infer that the use of lexical resources in the concept topic identification and query expansion processes does indeed prove beneficial.

Another contribution of our work is the development of a Web-based corpus of documents including sufficiently rich collections in multiple subject categories. This enabled us to cover a wide variety of topics wihtout the need to put topic restrictions on user queries. We also created a mapping between these subject categories from the Open Directory Project and the domains from WordNet Domains.

APPENDIX A

UTA-295 ODP CORPUS CATEGORIES

In this appendix, we list the categories from our UTA-295 ODP text corpus. The total size of the corpus is over 4.5 GB with 916,682 documents. There are 13 top level categories, which have a total of 295 categories within. Table A.1 lists the categories by top level category. These categories were obtained after the dividing and merging steps as explained in Chapter 3.

Table A.1: UTA-295 ODP Corpus Categories

Top-level Category	**Categories**
Arts	*Arts_Animation* *Arts_Animation_Anime* *Arts_Architecture* *Arts_Art_History* *Arts_Bodyart* *Arts_Comics* *Arts_Crafts* *Arts_Design* *Arts_Education* *Arts_Genres* *Arts_Illustration* *Arts_Literature* *Arts_Literature_Authors* *Arts_Literature_Genres* *Arts_Literature_World_Literature* *Arts_Movies* *Arts_Movies_Titles* *Arts_Music* *Arts_Music_Bands_Artists* *Arts_Music_Composition* *Arts_Music_Instruments* *Arts_Music_Styles* *Arts_Music_Vocal* *Arts_Online_Writing* *Arts_Other* *Arts_People* *Arts_Performing_Arts* *Arts_Performing_Arts_Acting* *Arts_Performing_Arts_Dance* *Arts_Performing_Arts_Theatre*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Arts_Photography *Arts_Radio* *Arts_Television* *Arts_Television_Programs* *Arts_Visual_Arts* *Arts_Visual_Arts_Painting* *Arts_Writers_Resources*
Business	*Business_Accounting* *Business_Aerospace_Defense* *Business_Agriculture_Forestry* *Business_Agriculture_Forestry_Livestock* *Business_Arts_Entertainment* *Business_Arts_Entertainment_Music* *Business_Arts_Entertainment_Photography* *Business_Automotive* *Business_Biotechnology_Pharmaceuticals* *Business_Business_Services* *Business_Business_Services_Communications* *Business_Business_Services_Design* *Business_Business_Services_Fire_Security* *Business_Chemicals* *Business_Construction_Maintenance* *Business_Construction_Maintenance_Design* *Business_Construction_Maintenance_Materials_Supplies* *Business_Consumer_Goods_Services* *Business_Consumer_Goods_Services_Clothing* *Business_Consumer_Goods_Services_Home_Garden* *Business_Consumer_Goods_Services_Sporting_Goods* *Business_E-Commerce* *Business_Electronics_Electrical* *Business_Employment* *Business_Energy_Environment* *Business_Financial_Services* *Business_Financial_Services_Banking_Services* *Business_Financial_Services_Insurance* *Business_Food_Related_Products* *Business_Healthcare* *Business_Hospitality*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Business_Human_Resources *Business_Industrial_Goods_Services* *Business_Industrial_Goods_Services_Casting_Molding_Machining* *Business_Industrial_Goods_Services_Fluid_Handling* *Business_Industrial_Goods_Services_Industrial_Supply* *Business_Industrial_Goods_Services_Machinery_Tools* *Business_Information_Technology* *Business_International_Business_Trade* *Business_Investing* *Business_Management* *Business_Marketing_Advertising* *Business_Opportunities* *Business_Other* *Business_Publishing_Printing* *Business_Real_Estate* *Business_Retail_Trade* *Business_Small_Business* *Business_Telecommunications* *Business_Textiles_Nonwovens* *Business_Textiles_Nonwovens_Textiles* *Business_Transportation_Logistics*
Computers	*Computers_Artificial_Intelligence* *Computers_CAD_CAM* *Computers_Computer_Science* *Computers_Consultants* *Computers_Data_Communications* *Computers_Data_Formats* *Computers_Education* *Computers_Graphics* *Computers_Hardware* *Computers_Internet* *Computers_Internet_On_the_Web* *Computers_Internet_RFCs* *Computers_Internet_Web_Design_Development* *Computers_Multimedia* *Computers_Other* *Computers_Programming*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Computers_Programming_Languages *Computers_Robotics* *Computers_Security* *Computers_Software* *Computers_Software_Databases* *Computers_Software_Graphics* *Computers_Software_Internet* *Computers_Software_Operating_Systems* *Computers_Software_Shareware* *Computers_Systems*
Games	*Games_Board_Games* *Games_Card_Games* *Games_Gambling* *Games_Miniatures* *Games_Online* *Games_Other* *Games_Roleplaying* *Games_Video_Games* *Games_Video_Games_Action* *Games_Video_Games_Adventure* *Games_Video_Games_Roleplaying* *Games_Video_Games_Shooter* *Games_Video_Games_Simulation* *Games_Video_Games_Sports* *Games_Video_Games_Strategy*
Health	*Health_Addictions* *Health_Alternative* *Health_Animal* *Health_Conditions_Diseases* *Health_Conditions_Diseases_Neurological_Disorders* *Health_Fitness* *Health_Medicine* *Health_Medicine_Medical_Specialties* *Health_Mental_Health* *Health_Nursing* *Health_Other* *Health_Pharmacy*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Health_Professions *Health_Public_Health_Safety* *Health_Reproductive_Health*
Home	*Home_Consumer_Information* *Home_Cooking* *Home_Cooking_Soups_Stews* *Home_Cooking_World_Cuisines* *Home_Family* *Home_Gardening* *Home_Other*
News	*News_Media* *News_Newspapers* *News_Other*
Recreation	*Recreation_Antiques* *Recreation_Autos* *Recreation_Aviation* *Recreation_Birding* *Recreation_Boating* *Recreation_Camps* *Recreation_Climbing* *Recreation_Collecting* *Recreation_Food* *Recreation_Guns* *Recreation_Humor* *Recreation_Living_History* *Recreation_Models* *Recreation_Motorcycles* *Recreation_Other* *Recreation_Outdoors* *Recreation_Outdoors_Camping* *Recreation_Outdoors_Hunting* *Recreation_Pets* *Recreation_Pets_Cats* *Recreation_Pets_Dogs* *Recreation_Radio* *Recreation_Roads_Highways*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Recreation_Scouting *Recreation_Trains_Railroads* *Recreation_Travel*
Reference	*Reference_Dictionaries* *Reference_Education* *Reference_Education_Colleges_Universities* *Reference_Education_K_through_* *Reference_Encyclopedias* *Reference_Knowledge_Management* *Reference_Libraries* *Reference_Museums* *Reference_Other*
Science	*Science_Agriculture* *Science_Astronomy* *Science_Biology* *Science_Biology_Flora_Fauna* *Science_Chemistry* *Science_Earth_Sciences* *Science_Environment* *Science_Instruments_Supplies* *Science_Math* *Science_Other* *Science_Physics* *Science_Social_Sciences* *Science_Social_Sciences_Archaeology* *Science_Social_Sciences_Linguistics* *Science_Social_Sciences_Psychology* *Science_Technology* *Science_Technology_Energy*
Shopping	*Shopping_Antiques_Collectibles* *Shopping_Children* *Shopping_Clothing* *Shopping_Consumer_Electronics* *Shopping_Crafts* *Shopping_Crafts_Supplies* *Shopping_Entertainment*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Shopping_Ethnic_Regional *Shopping_Flowers* *Shopping_Food* *Shopping_General_Merchandise* *Shopping_Gifts* *Shopping_Health* *Shopping_Home_Garden* *Shopping_Home_Garden_Accessories* *Shopping_Home_Garden_Furniture* *Shopping_Jewelry* *Shopping_Music* *Shopping_Niche* *Shopping_Other* *Shopping_Pets* *Shopping_Publications* *Shopping_Recreation* *Shopping_Sports* *Shopping_Toys_Games* *Shopping_Vehicles* *Shopping_Visual_Arts*
Society	*Society_Activism* *Society_Crime* *Society_Death* *Society_Disabled* *Society_Ethnicity* *Society_Folklore* *Society_Gay_Lesbian_Bisexual* *Society_Genealogy* *Society_Government* *Society_History* *Society_History_By_Time_Period* *Society_Holidays* *Society_Issues* *Society_Issues_Environment* *Society_Issues_Health* *Society_Issues_Warfare_Conflict* *Society_Law* *Society_Law_Services*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Society_Military *Society_Organizations* *Society_Organizations_Fraternal* *Society_Organizations_Student* *Society_Other* *Society_Paranormal* *Society_Philanthropy* *Society_Philosophy* *Society_Politics* *Society_Relationships* *Society_Religion_Spirituality* *Society_Religion_Spirituality_Buddhism* *Society_Religion_Spirituality_Christianity* *Society_Religion_Spirituality_Esoteric_Occult* *Society_Religion_Spirituality_Islam* *Society_Religion_Spirituality_Judaism* *Society_Religion_Spirituality_Pagan* *Society_Religion_Spirituality_Yoga* *Society_Subcultures* *Society_Transgendered*
Sports	*Sports_Baseball* *Sports_Basketball* *Sports_Cricket* *Sports_Cycling* *Sports_Equestrian* *Sports_Equestrian_Breeds* *Sports_Football* *Sports_Golf* *Sports_Hockey* *Sports_Martial_Arts* *Sports_Motorsports* *Sports_Other* *Sports_Running* *Sports_Skating* *Sports_Soccer* *Sports_Soccer_UEFA* *Sports_Softball* *Sports_Tennis*

Table A.1: *(continued)*

Top-level Category	**Categories**
	Sports_Track_Field *Sports_Volleyball* *Sports_Water_Sports* *Sports_Winter_Sports* *Sports_Wrestling*

APPENDIX B

WORDNET DOMAINS 3.1 - UTA-295 ODP CORPUS MAPPING

The final mapping between each WordNet (WN) Domain 3.1 [14] and the relevant categories from our UTA-295 ODP corpus, that was used in our research, is shown in Table B.1.

Table B.1: WN Domain - UTA-295 Category Mapping

WordNet Domain	**UTA-295 ODP Corpus Categories**
humanities	*Arts_Other* *Society_Religion_Spirituality_Christianity* *Society_Religion_Spirituality_Islam*
history	*Society_History* *Arts_Art_History* *Society_History_By_Time_Period* *Recreation_Living_History*
archaeology	*Science_Social_Sciences_Archaeology* *Society_Ethnicity*
heraldry	*Society_Genealogy* *Recreation_Living_History* *Arts_Music_Composition* *Arts_Literature_World_Literature*
linguistics	*Science_Social_Sciences_Linguistics* *Arts_Education* *Society_Philosophy*
grammar	*Science_Social_Sciences_Linguistics* *Arts_Education* *Reference_Dictionaries* *Shopping_General_Merchandise*
literature	*Arts_Literature* *Society_Folklore* *Reference_Other* *Arts_Other* *Arts_Literature_Authors* *Arts_Literature_World_Literature* *Arts_Literature_Genres*
philology	*Science_Social_Sciences_Linguistics* *Reference_Other*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Arts_Other
philosophy	*Society_Philosophy* *Arts_Writers_Resources* *Arts_Literature_Genres*
psychology	*Science_Social_Sciences_Psychology* *Health_Mental_Health*
psychoanalysis	*Science_Social_Sciences_Psychology* *Health_Mental_Health* *Society_Philosophy*
art	*Arts_Other* *Business_Arts_Entertainment* *Arts_Performing_Arts* *Arts_Illustration* *Arts_Movies_Titles* *Arts_Music_Bands_Artists* *Arts_Literature_Authors* *Arts_Music* *Arts_Literature_World_Literature* *Arts_Music_Styles* *Arts_Music_Composition* *Arts_People* *Arts_Television_Programs* *Arts_Literature* *Arts_Literature_Genres* *Arts_Performing_Arts_Acting* *Arts_Movies* *Science_Earth_Sciences* *Arts_Music_Instruments* *Arts_Animation* *Arts_Animation_Anime* *Arts_Visual_Arts* *Arts_Online_Writing* *Arts_Comics* *Arts_Crafts* *Arts_Writers_Resources*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Arts_Architecture *Sports_Martial_Arts* *Arts_Performing_Arts_Dance* *Arts_Art_History* *Arts_Performing_Arts_Theatre* *Arts_Photography* *Shopping_Visual_Arts* *Arts_Television* *Arts_Radio* *Business_Arts_Entertainment_Music* *Business_Arts_Entertainment_Photography* *Computers_Artificial_Intelligence* *Arts_Design* *Arts_Education* *Arts_Music_Vocal* *Arts_Visual_Arts_Painting* *Arts_Genres* *Arts_Bodyart*
dance	*Arts_Performing_Arts* *Arts_Music* *Business_Arts_Entertainment* *Arts_Performing_Arts_Dance* *Arts_Performing_Arts_Theatre* *Shopping_Niche*
drawing	*Arts_Visual_Arts* *Arts_Animation_Anime* *Shopping_Recreation* *Business_Industrial_Goods_Services_Machinery_Tools* *Arts_Architecture*
graphic_arts	*Science_Other* *Arts_Visual_Arts*
philately	*Recreation_Collecting* *Arts_Movies_Titles* *Arts_Music*
painting	*Arts_Visual_Arts*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Arts_Crafts *Arts_Illustration* *Arts_Bodyart* *Arts_Visual_Arts_Painting*
music	*Arts_Music* *Business_Arts_Entertainment* *Arts_Music_Bands_Artists* *Arts_Music_Styles* *Arts_Music_Composition* *Arts_Music_Instruments* *Business_Arts_Entertainment_Music* *Shopping_Music* *Arts_Music_Vocal*
photography	*Arts_Photography* *Business_Arts_Entertainment* *Business_Arts_Entertainment_Photography*
plastic_arts	*Arts_Visual_Arts* *Business_Chemicals* *Business_Consumer_Goods_Services* *Shopping_Home_Garden_Accessories*
jewellery	*Business_Consumer_Goods_Services* *Arts_Crafts* *Society_Ethnicity*
numismatics	*Recreation_Collecting* *Reference_Museums* *Business_Opportunities*
sculpture	*Arts_Visual_Arts* *Science_Earth_Sciences* *Shopping_Jewelry*
theatre	*Arts_Movies* *Arts_Performing_Arts* *Arts_Performing_Arts_Theatre* *Science_Social_Sciences_Psychology*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Arts_Music_Composition *Arts_Radio* *Arts_Literature_Genres*
cinema	*Arts_Movies* *Business_Arts_Entertainment_Photography* *Arts_Photography*
paranormal	*Society_Paranormal* *Science_Social_Sciences_Psychology*
occultism	*Society_Religion_Spirituality*
astrology	*Society_Religion_Spirituality* *Society_Paranormal*
religion	*Society_Religion_Spirituality* *Society_Religion_Spirituality_Christianity* *Society_Religion_Spirituality_Esoteric_Occult* *Society_Religion_Spirituality_Buddhism* *Society_Religion_Spirituality_Islam* *Society_Religion_Spirituality_Pagan* *Society_Religion_Spirituality_Judaism* *Society_Religion_Spirituality_Yoga*
theology	*Society_Religion_Spirituality* *Reference_Libraries* *Science_Other* *Society_Religion_Spirituality_Christianity*
roman_catholic	*Society_Religion_Spirituality_Christianity* *Reference_Education_Colleges_Universities* *Sports_Baseball* *Sports_Track_Field*
mythology	*Arts_Literature* *Society_Folklore* *Society_Religion_Spirituality*
free_time	*Recreation_Pets_Dogs* *Recreation_Outdoors*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Recreation_Travel *Recreation_Autos* *Recreation_Humor* *Recreation_Collecting* *Recreation_Pets* *Shopping_Recreation* *Recreation_Scouting* *Recreation_Food* *Recreation_Aviation* *Recreation_Pets_Cats* *Recreation_Radio* *Recreation_Birding* *Recreation_Boating* *Recreation_Outdoors_Hunting* *Recreation_Other* *Recreation_Motorcycles* *Recreation_Outdoors_Camping* *Recreation_Roads_Highways* *Recreation_Living_History* *Recreation_Climbing* *Recreation_Guns* *Recreation_Antiques* *Recreation_Models* *Recreation_Trains_Railroads* *Recreation_Camps* *Sports_Other* *Sports_Basketball* *Sports_Soccer_UEFA* *Sports_Football* *Shopping_Sports* *Sports_Baseball* *Sports_Hockey* *Sports_Soccer* *Business_Transportation_Logistics* *Sports_Motorsports* *Sports_Martial_Arts* *Sports_Equestrian_Breeds* *Games_Video_Games_Sports* *Sports_Cycling*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Sports_Water_Sports *Sports_Golf* *Sports_Equestrian* *Sports_Running* *Sports_Tennis* *Sports_Winter_Sports* *Sports_Track_Field* *Sports_Volleyball* *Sports_Softball* *Sports_Wrestling* *Sports_Skating* *Sports_Cricket* *Games_Video_Games* *Games_Video_Games_Roleplaying* *Games_Video_Games_Shooter* *Games_Video_Games_Adventure* *Games_Roleplaying* *Games_Video_Games_Action* *Games_Video_Games_Strategy* *Games_Other* *Games_Board_Games* *Games_Video_Games_Simulation* *Games_Online* *Games_Gambling* *Games_Card_Games* *Games_Miniatures* *Computers_Internet*
radio+tv	*Arts_Radio* *Arts_Television* *Shopping_Antiques_Collectibles* *Home_Consumer_Information* *Business_Telecommunications*
play	*Games_Video_Games* *Games_Video_Games_Roleplaying* *Games_Video_Games_Shooter* *Games_Video_Games_Adventure* *Games_Roleplaying*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Games_Video_Games_Action *Games_Video_Games_Strategy* *Games_Video_Games_Sports* *Games_Other* *Games_Board_Games* *Games_Video_Games_Simulation* *Games_Online* *Games_Gambling* *Games_Card_Games* *Games_Miniatures*
betting	*Games_Gambling* *Games_Board_Games* *Sports_Golf* *Business_Arts_Entertainment*
card	*Games_Card_Games* *Games_Other* *Games_Gambling* *Arts_Crafts* *Society_Organizations* *Arts_Illustration* *Shopping_Sports*
chess	*Games_Board_Games* *Games_Video_Games* *Computers_Systems* *Shopping_Toys_Games*
sport	*Sports_Other* *Sports_Basketball* *Sports_Soccer_UEFA* *Sports_Football* *Shopping_Sports* *Sports_Baseball* *Sports_Hockey* *Sports_Soccer* *Business_Transportation_Logistics* *Sports_Motorsports*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Sports_Martial_Arts *Sports_Equestrian_Breeds* *Games_Video_Games_Sports* *Sports_Cycling* *Sports_Water_Sports* *Sports_Golf* *Sports_Equestrian* *Sports_Running* *Sports_Tennis* *Sports_Winter_Sports* *Sports_Track_Field* *Sports_Volleyball* *Sports_Softball* *Business_Consumer_Goods_Services_Sporting_Goods* *Sports_Wrestling* *Sports_Skating* *Sports_Cricket*
badminton	*Sports_Other* *Business_Consumer_Goods_Services_Sporting_Goods*
baseball	*Sports_Baseball* *Games_Video_Games* *Sports_Other* *Business_Industrial_Goods_Services*
basketball	*Sports_Basketball* *Games_Video_Games* *Sports_Other* *Reference_Education_Colleges_Universities*
cricket	*Sports_Cricket* *Sports_Other* *Business_Consumer_Goods_Services* *Games_Video_Games*
football	*Sports_Football* *Games_Video_Games* *Sports_Other*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
golf	*Sports_Golf* *Business_Consumer_Goods_Services* *Sports_Other* *Games_Video_Games*
rugby	*Sports_Football* *Games_Video_Games* *Sports_Other* *Sports_Soccer* *Sports_Soccer_UEFA*
soccer	*Sports_Soccer* *Games_Video_Games* *Sports_Other* *Sports_Soccer_UEFA*
table_tennis	*Sports_Other* *Business_Consumer_Goods_Services*
tennis	*Sports_Tennis* *Games_Video_Games* *Sports_Other* *Business_Consumer_Goods_Services*
volleyball	*Sports_Volleyball* *Games_Video_Games* *Business_Consumer_Goods_Services_Sporting_Goods*
cycling	*Sports_Cycling* *Business_Consumer_Goods_Services* *Shopping_Sports*
skating	*Sports_Skating* *Sports_Hockey* *Games_Video_Games* *Shopping_Sports*
skiing	*Sports_Winter_Sports* *Sports_Water_Sports* *Sports_Other*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
hockey	*Sports_Hockey* *Sports_Other* *Games_Other* *Business_Consumer_Goods_Services* *Shopping_Sports*
mountaineering	*Recreation_Climbing* *Recreation_Outdoors*
rowing	*Sports_Water_Sports* *Business_Consumer_Goods_Services* *Sports_Other*
swimming	*Sports_Water_Sports* *Business_Consumer_Goods_Services*
sub	*Recreation_Outdoors* *Business_Consumer_Goods_Services_Sporting_Goods*
diving	*Sports_Water_Sports* *Recreation_Outdoors* *Business_Consumer_Goods_Services* *Sports_Other*
racing	*Sports_Motorsports* *Games_Video_Games* *Sports_Other* *Business_Automotive*
athletics	*Sports_Running* *Sports_Track_Field*
wrestling	*Sports_Wrestling* *Sports_Other*
boxing	*Sports_Other* *Shopping_Sports*
fencing	*Sports_Other* *Business_Consumer_Goods_Services*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
archery	*Sports_Other* *Business_Consumer_Goods_Services*
fishing	*Recreation_Outdoors* *Business_Consumer_Goods_Services* *Recreation_Boating*
hunting	*Recreation_Outdoors* *Recreation_Outdoors_Hunting*
bowling	*Sports_Other* *Business_Consumer_Goods_Services* *Games_Video_Games* *Business_Consumer_Goods_Services_Sporting_Goods* *Shopping_Sports*
applied_science	*Science_Biology_Flora_Fauna* *Science_Technology* *Science_Other* *Science_Technology_Energy* *Science_Agriculture* *Computers_Computer_Science* *Science_Instruments_Supplies*
agriculture	*Science_Agriculture* *Business_Agriculture_Forestry* *Business_Agriculture_Forestry_Livestock*
animal_husbandry	*Science_Other* *Health_Animal*
veterinary	*Health_Animal* *Business_Healthcare* *Business_Biotechnology_Pharmaceuticals*
food	*Home_Cooking* *Shopping_Food* *Recreation_Food* *Business_Food_Related_Products*
gastronomy	*Home_Cooking*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Home_Family
home	*Home_Other* *Home_Cooking* *Shopping_Home_Garden* *Home_Family* *Home_Gardening* *Home_Consumer_Information* *Home_Cooking_Soups_Stews* *Home_Cooking_World_Cuisines* *Shopping_Home_Garden_Furniture* *Shopping_Home_Garden_Accessories* *Business_Consumer_Goods_Services_Home_Garden*
architecture	*Arts_Architecture* *Business_Construction_Maintenance* *Business_Construction_Maintenance_Materials_Supplies*
town_planning	*Science_Social_Sciences* *Business_Construction_Maintenance* *Arts_Architecture*
building_industry	*Business_Construction_Maintenance*
furniture	*Business_Consumer_Goods_Services* *Arts_Crafts* *Shopping_Home_Garden_Furniture*
computer_science	*Computers_Computer_Science* *Computers_Internet_RFCs* *Computers_Programming_Languages* *Computers_Software* *Computers_Internet_Web_Design_Development* *Computers_Internet* *Computers_Internet_On_the_Web* *Computers_Software_Operating_Systems* *Computers_Other* *Computers_Programming* *Computers_Hardware* *Computers_Software_Internet*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Computers_Data_Formats *Computers_Systems* *Computers_Security* *Computers_Software_Shareware* *Computers_Multimedia* *Computers_Software_Graphics* *Computers_Software_Databases* *Computers_Artificial_Intelligence* *Computers_Data_Communications* *Computers_Consultants* *Computers_Graphics* *Computers_Education* *Computers_Robotics* *Computers_CAD_CAM*
engineering	*Science_Technology* *Business_Industrial_Goods_Services*
mechanics	*Science_Physics* *Business_Industrial_Goods_Services*
astronautics	*Business_Aerospace_Defense* *Science_Technology* *Recreation_Aviation* *Science_Astronomy*
electrotechnology	*Science_Technology* *Business_Electronics_Electrical* *Recreation_Collecting* *Business_Consumer_Goods_Services*
hydraulics	*Business_Industrial_Goods_Services* *Business_Construction_Maintenance* *Science_Chemistry* *Science_Earth_Sciences* *Business_Transportation_Logistics*
telecommunication	*Business_Telecommunications*
post	*Business_Transportation_Logistics*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Society_Government *Reference_Other* *Games_Roleplaying*
telegraphy	*Business_Telecommunications* *Business_Electronics_Electrical* *Computers_Data_Communications*
telephony	*Business_Telecommunications*
medicine	*Health_Medicine* *Health_Alternative* *Health_Medicine_Medical_Specialties*
dentistry	*Health_Other* *Health_Conditions_Diseases* *Business_Industrial_Goods_Services_Machinery_Tools*
pharmacy	*Health_Pharmacy* *Business_Biotechnology_Pharmaceuticals*
psychiatry	*Health_Medicine* *Society_Issues* *Society_Crime*
radiology	*Health_Medicine* *Health_Medicine_Medical_Specialties* *Health_Nursing*
surgery	*Health_Medicine* *Health_Other* *Science_Physics* *Health_Reproductive_Health* *Business_Biotechnology_Pharmaceuticals*
pure_science	*Science_Math* *Science_Biology_Flora_Fauna* *Science_Biology* *Science_Physics* *Science_Environment* *Science_Earth_Sciences*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Science_Other *Science_Chemistry* *Science_Astronomy* *Science_Technology_Energy* *Business_Industrial_Goods_Services*
astronomy	*Science_Astronomy* *Science_Physics* *Science_Technology*
biology	*Science_Biology* *Science_Biology_Flora_Fauna*
biochemistry	*Science_Biology* *Arts_Television* *Arts_Music_Bands_Artists* *Society_Philosophy*
anatomy	*Health_Medicine* *Health_Other* *Health_Reproductive_Health*
physiology	*Science_Biology*
genetics	*Science_Biology* *Society_Issues* *Business_Biotechnology_Pharmaceuticals*
animals	*Science_Biology* *Science_Biology_Flora_Fauna* *Business_Textiles_Nonwovens_Textiles*
entomology	*Science_Biology* *Science_Biology_Flora_Fauna*
plants	*Science_Biology* *Science_Agriculture* *Science_Biology_Flora_Fauna*
environment	*Science_Environment* *Society_Issues*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Society_Issues_Environment *Business_Energy_Environment*
chemistry	*Science_Chemistry* *Science_Earth_Sciences*
earth	*Science_Earth_Sciences*
geology	*Science_Earth_Sciences*
meteorology	*Science_Earth_Sciences*
oceanography	*Science_Earth_Sciences* *Reference_Education_Colleges_Universities*
paleontology	*Science_Earth_Sciences* *Science_Social_Sciences_Archaeology* *Science_Biology_Flora_Fauna*
geography	*Science_Social_Sciences*
topography	*Science_Earth_Sciences* *Business_Industrial_Goods_Services_Machinery_Tools*
mathematics	*Science_Math*
geometry	*Science_Math* *Science_Physics*
statistics	*Science_Math* *Science_Physics*
physics	*Science_Physics*
acoustics	*Science_Technology* *Business_Consumer_Goods_Services* *Business_Energy_Environment* *Science_Other*
atomic_physic	*Science_Physics* *Science_Technology*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Business_Energy_Environment
electricity	*Science_Technology* *Business_Energy_Environment*
electronics	*Science_Technology* *Business_Electronics_Electrical* *Shopping_Consumer_Electronics*
gas	*Science_Physics*
optics	*Science_Physics* *Business_Industrial_Goods_Services*
social_science	*Science_Social_Sciences* *Science_Social_Sciences_Linguistics* *Science_Social_Sciences_Archaeology* *Science_Social_Sciences_Psychology*
anthropology	*Science_Social_Sciences* *Reference_Museums*
ethnology	*Science_Social_Sciences* *Reference_Museums* *Society_History* *Recreation_Antiques*
folklore	*Society_Folklore* *Arts_Comics*
health	*Health_Other* *Health_Conditions_Diseases* *Health_Mental_Health* *Shopping_Health* *Society_Issues_Health* *Health_Alternative* *Health_Medicine* *Health_Pharmacy* *Health_Medicine_Medical_Specialties* *Health_Conditions_Diseases_Neurological_Disorders* *Business_Healthcare*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Health_Animal *Health_Addictions* *Health_Public_Health_Safety* *Health_Reproductive_Health* *Health_Nursing* *Health_Professions* *Health_Fitness*
body_care	*Business_Consumer_Goods_Services* *Health_Other*
military	*Society_Military*
pedagogy	*Society_Issues* *Reference_Education* *Arts_Education*
school	*Reference_Education*
university	*Reference_Education*
publishing	*Business_Publishing_Printing*
sociology	*Science_Social_Sciences* *Society_Issues*
artisanship	*Arts_Crafts* *Recreation_Models* *Shopping_Other* *Arts_Visual_Arts* *Science_Social_Sciences* *Computers_CAD_CAM* *Business_Industrial_Goods_Services*
commerce	*Business_E-Commerce* *Business_International_Business_Trade* *Business_Retail_Trade* *Society_Activism*
industry	*Business_Industrial_Goods_Services* *Business_Small_Business*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Business_Textiles_Nonwovens *Business_Textiles_Nonwovens_Textiles* *Business* *Business_Financial_Services* *Business_Investing* *Business_Real_Estate* *Business_Other*
transport	*Business_Transportation_Logistics* *Shopping_Recreation*
aviation	*Recreation_Aviation*
vehicles	*Recreation_Autos* *Shopping_Vehicles*
nautical	*Business_Transportation_Logistics* *Science_Earth_Sciences*
railway	*Business_Transportation_Logistics* *Recreation_Trains_Railroads*
economy	*Science_Social_Sciences* *Business_Financial_Services*
enterprise	*Business_Management*
book_keeping	*Business_Business_Services*
finance	*Business_Financial_Services* *Home_Other* *Society_Philanthropy*
banking	*Business_Financial_Services* *Business_Investing* *Business_Financial_Services_Banking_Services*
exchange	*Business_Investing*
money	*Home_Other* *Business_Financial_Services*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Business_Investing *Shopping_Antiques_Collectibles* *Recreation_Antiques*
insurance	*Business_Financial_Services* *Home_Other* *Society_Government* *Business_Financial_Services_Insurance*
tax	*Business_Accounting* *Business_Financial_Services* *Home_Other* *Society_Issues* *Business_Management* *Reference_Libraries*
administration	*Business_Other* *Science_Social_Sciences*
law	*Society_Law* *Society_Law_Services*
politics	*Society_Politics* *Society_Other*
diplomacy	*Society_Politics*
tourism	*Recreation_Travel* *Business_Agriculture_Forestry*
fashion	*Arts_Design* *Business_Consumer_Goods_Services* *Business_Arts_Entertainment* *Games_Video_Games_Simulation*
sexuality	*Society_Other*
time_period	*Society_History* *Society_History_By_Time_Period*
person	*Society_Other*

Table B.1: *(continued)*

WordNet Domain	**UTA-295 ODP Corpus Categories**
	Home_Other *News_Other*
metrology	*Science_Technology*
psychological_features	*Science_Social_Sciences* *Games_Video_Games_Roleplaying* *Recreation_Boating* *Shopping_Other*

APPENDIX C

STANDARD QUERIES

In this appendix, we list the standard queries that we presented to the users. These one word queries were obtained from the Senseval 3 competition [43], the TWA Sense Tagged Data [44, 45], and a research publication [46].

Table C.1 Standard Queries

act	activate	add	animal
appear	argument	arm	artifact
ask	atmosphere	attribute	audience
bank	bass	begin	body
climb	cognition	communication	crane
decide	degree	difference	different
difficulty	disc	eat	encounter
event	expect	express	feeling
food	group	hear	hot
image	important	interest	judgment
location	lose	mean	miss
motion	motive	note	object
operate	organization	palm	paper
party	performance	person	phenomenon
plan	plant	play	possession
process	produce	provide	quantity
receive	relation	remain	rule
shape	shelter	simple	smell
solid	sort	source	state
substance	suspend	talk	tank
time	treat	use	wash
watch	win	write	

APPENDIX D

USER QUERIES

In this appendix, we list the queries that the users entered and reported during the evaluations. These are the queries used in addition to those mentioned in Appendix C.

Table D.1: User Queries

acetaminophen	add	adobe professional
apple	arlington	automatic
axon	baby	baseball
bat	bed bugs	bell
binary	birds	blackberry
bombay	card	cell
cell line	chat	chick
choclate	class	close
cobol	constructor	cplusplus
crane	cricket	csharp
cystic fibrosis	delegate	dielectric
difference cd dvd	donut	einstein
enterprise	exercise	expression
extern	ferrari	function
game	ghalib	gold bonding
google	hand	harry potter
hash	histocompatibility	hit
home	honda	html
imaging	index	instance
integer	interview	interview preparation
java	jfk	juice
jump	langerhans cell	language
laptop	lara	linklist
lion	lipid membrane	loop
map new york	matrix	medical
medicine	metro ethernet forum	microsoft
mining	mitotic spindle	mole
mould	mouse	net
netbeans	north gate wa	nucleus
opera	operation	pacemaker
paris	park	pattern
pentagon	perl	pga
plane	pointer	program
protein adsorption	protocol	query
queue	recursion	register
regulatory t cells	ruby	rule
saline	scope	search engines

Table D.1: User Queries *(continued)*

server	shoes	skin
soap	solid	statements
stock	sun	tea
test	toll like receptors	transplantation
tree structure	variable	virus
vivian richards	volley	west indies
wilms tumor	window	www
xpert	yahoo	

REFERENCES

[1] C. Stokoe, “Automated Word Sense Disambiguation for Web Information Retrieval,” Ph.D. dissertation, University of Sunderland, Sunderland, England, July 2004.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, “Indexing by Latent Semantic Analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] Search Engine Watch. Incisive Interactive Marketing LLC. [Online]. Available: http://searchenginewatch.com/

[4] Google search engine. Google. [Online]. Available: http://www.google.com/

[5] BizRate Shopping Search. Shopzilla Inc. [Online]. Available: http://www.bizrate.com/

[6] KidsClick! Web search for kids by librarians. http://www.kidsclick.org/. [Online]. Available: http://www.kidsclick.org/

[7] HotBot. Lycos Inc. [Online]. Available: http://www.hotbot.com/

[8] Open Directory Project. Netscape Communications Corp. Mountain View, CA. [Online]. Available: http://dmoz.org/

[9] CiteSeer.IST. NEC and Pennsylvania State University. [Online]. Available: http://citeseer.ist.psu.edu/

[10] Ask.com. IAC Search and Media. [Online]. Available: http://www.ask.com/

[11] B. J. Jansen, A. Spink, and J. Pedersen, “A Temporal Comparison of AltaVista Web Searching,” *Journal of the American Society for Information Science and Technology*, vol. 56, no. 6, pp. 559–570, 2005.

[12] B. J. Jansen and A. Spink, “How are we searching the World Wide Web? A comparison of nine search engine transaction logs,” *Information Processing and Management*, vol. 42, no. 6, pp. 248–263, 2006.

[13] WordNet: a lexical database for the English language. Princeton University. Princeton, NJ. [Online]. Available: http://wordnet.princeton.edu/

[14] L. Bentivogli, P. Forner, B. Magnini, and E. Pianta, “Revising WordNet Domains Hierarchy: Semantics, Coverage, and Balancing,” in *Proceedings of COLING 2004 Workshop on Multilingual Linguistic Resources*, Aug. 28, 2004, pp. 101–108.

[15] D. Stenmark, “Query Expansion on a Corporate Intranet: Using LSI to Increase Precision in Explorative Search,” in *Proceedings of the 38th Hawaii International Conference on System Sciences - Track 4*, 2005, p. 101.c.

[16] H. Schütze, *Ambiguity Resolution in Language Learning.* Stanford, CA: CSLI Publications, 1997.

[17] Infomap-NLP Software. [Online]. Available: http://infomap.stanford.edu/

[18] S. Brin and L. Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine,” in *Proceedings of the Seventh World-Wide Web Conference*, Apr. 1998, pp. 107–117.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank Citation Ranking: Bringing Order to the Web,” Stanford Digital Library Technologies Project, Tech. Rep., 1998.

[20] M. S. Khan and S. Khor, “Enhanced Web Document Retrieval using Automatic Query Expansion,” *Journal of the American Society for Information Science and Technology*, vol. 55, pp. 29–40, 2004.

[21] E. N. Efthimiadis, “Query Expansion,” *Annual Review of Information Systems and Technology*, vol. 31, pp. 121–187, 1996.

[22] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* Harlow, England: Addison-Wesley, 1999.

[23] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, pp. 613–620, 1975.

[24] D. Widdows, *Geometry and Meaning.* Stanford, CA: CSLI Publications, 2004.

[25] Text REtrieval Conference (TREC). NIST and ARDA. [Online]. Available: http://trec.nist.gov/

[26] Reuters Corpus, Volume 1, English language, 1996-08-20 to 1997-08-19 (Release date 2000-11-03, Format version 1, correction level 0). Reuters Ltd. [Online]. Available: http://about.reuters.com/researchandstandards/corpus/index.asp

[27] BioMed Central Corpus. BioMed Central Ltd. London, England. [Online]. Available: http://www.biomedcentral.com/info/about/datamining/

[28] D. Davidov, E. Gabrilovich, and S. Markovitch, "Parameterized Generation of Labeled Datasets for Text Categorization Based on a Hierarchical Directory," in *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, 2004, pp. 250–257.

[29] W. A. Gale, K. W. Church, and D. Yarowsky, "One Sense per Discourse," in *Proceedings of the workshop on Speech and Natural Language*, 1992, pp. 233–237.

[30] D. Yarowsky, "One Sense per Collocation," in *Proceedings of the workshop on Human Language Technology*, vol. 8, 1993, pp. 266–271.

[31] D. Martinez and E. Agirre, "One Sense per Collocation and Genre/Topic Variations," in *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 2000, pp. 266–271.

[32] B. Magnini, C. Strapparava, G. Pezzulo, and A. Gliozzo, "The Role of Domain Information in Word Sense Disambiguation," *Natural Language Engineering*, vol. 8, no. 4, pp. 359–373, 2002.

[33] Northern Light. Northern Light Group LLC. Cambridge, MA. [Online]. Available: http://www.northernlight.com/

[34] M. F. Krellenstein, "Method and Apparatus for Searching a Database of Records," U.S. Patent 5 924 090, July 13, 1999.

[35] S. T. Dumais, D. D. Lewis, and F. Sebastiani, "Report on the workshop on Operational Text Classification Systems," SIGIR Forum, pp. 68–71, 2002.

[36] R. Ozcan, "Concept-based Information Access," Master's thesis, The University of Texas at Arlington, Arlington, TX, 2004.

[37] R. Ozcan and Y. A. Aslandogan, "Concept-based Information Access," in *Proceedings of the International Conference on Information Technology: Coding and Computing*, vol. 1, Apr. 4–6, 2005, pp. 794–799.

[38] HTMLParser. [Online]. Available: http://htmlparser.sourceforge.net/

[39] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, Apr. 2004.

[40] C. Santamaría, J. Gonzalo, and F. Verdejo, "Automatic Association of Web Directories to Word Senses," *Computational Linguistics - Special Issue on the Web as Corpus*, vol. 29, no. 3, pp. 485–502, Sept. 2003.

[41] Java WordNet Library. [Online]. Available: http://jwordnet.sourceforge.net/

[42] T. Rose, M. Stevenson, and M. Whitehead, "The Reuters Corpus Volume 1 - from Yesterday's News to Tomorrow's Language Resources," in *Proceedings of the Third International Conference on Language Resources and Evaluation*, May 29–31, 2002. [Online]. Available: http://about.reuters.com/researchandstandards/corpus/LREC_camera_ready.pdf

[43] Senseval: Evaluation exercises for Word Sense Disambiguation. The Association for Computational Linguistics. [Online]. Available: http://www.senseval.org/

[44] TWA Sense tagged data. Rada Mihalcea and Li Yang, University of North Texas. [Online]. Available: http://www.cs.unt.edu/~rada/downloads.html

[45] R. Mihalcea, "The Role of Non-Ambiguous Words in Natural Language Disambiguation," in *Proceedings of the Conference on Recent Advances in Natural Language Processing*, 2003.

[46] S.-B. Kim, H.-C. Seo, and H.-C. Rim, "Information retrieval using word senses: root sense tagging approach," in *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, 2004, pp. 258–265.

BIOGRAPHICAL INFORMATION

Rahul Rajiv Joshi received the B.E. degree in electronics engineering from University of Mumbai, India in 2001 and the M.S. degree in computer science and engineering from The University of Texas at Arlington in 2006. He worked as a Software Engineer at Patni Computer Systems from 2001 to 2003 and as a Software Developer in Ayoka Systems Engineering in summer of 2005. He is a member of Tau Beta Pi and Upsilon Pi Epsilon. His current research interests include algorithms, databases, data mining, and information retrieval.