

THE PENNSYLVANIA STATE UNIVERSITY  
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AN ANALYSIS OF THE ACCURACY OF A SEARCH ENGINE RANKING ALGORITHM  
FOR META-SEARCH ENGINES USING THE SUMMARY SCHEMAS MODEL

KATHRYN E. BECHTOLD

Fall 2000

A thesis  
submitted in partial fulfillment  
of the requirements  
for a baccalaureate degree  
in Computer Science  
with honors in Computer Science And Engineering

Approved: \_\_\_\_\_ Date: \_\_\_\_\_

Dr. A. R. Hurson  
Thesis Supervisor

\_\_\_\_\_ Date: \_\_\_\_\_

Dr. J. Hannan  
Honors Advisor

## Abstract

The Summary Schemas Model (SSM) was originally developed as a way to enhance the query capabilities of multidatabase systems by recognizing the semantic content of query terms. It has been proposed that the application domain of this model be extended to include the World Wide Web, which shares some characteristics of multidatabases. Specifically, if a meta-search engine could use SSM to semantically identify query terms, it could select a set of other search engines to query that tend to provide the most relevant results for queries in the given subject area. An algorithm called Qsearch has been developed to implement search engine selection based on the search engines' profiles, user feedback, and by providing search engine rankings supported by a Summary Schemas Model hierarchy. An evaluation of Qsearch's accuracy and learning ability is presented to compare Qsearch's search engine rankings with a baseline of intuitive, user-centered rankings. The strengths and limitations of Qsearch are discussed, and future research directions are presented.

## Table of Contents

ABSTRACT	i
TABLE OF CONTENTS	ii
LIST OF FIGURES	iv
LIST OF EQUATIONS	v
1 INTRODUCTION	1
2 BACKGROUND	3
2.1 MULTIDATABASES	3
2.2 THE SUMMARY SCHEMAS MODEL	5
2.3 WORLD WIDE WEB SEARCH ENGINES	9
2.3.1 <i>Robot-based Search Engines</i>	10
2.3.2 <i>Directory-based Search Engines</i>	10
2.3.3 <i>Meta-search Engines</i>	11
2.4 MOTIVATIONS FOR USING THE SSM IN SEARCH ENGINE SELECTION	12
2.5 CONCLUSIONS	12
3 IMPLEMENTATION: QSEARCH	14
3.1 USER INTERFACE	14
3.2 SSM HIERARCHICAL STRUCTURE	15
3.3 META-TABLE	18
3.4 CONCLUSIONS	20
4 EVALUATION METHODS	21
4.1 SEARCH ENGINE AND QUERY CHOICES	21
4.2 INITIAL RUN	22

4.3 USER FEEDBACK	23
4.4 POST-USER FEEDBACK RUN	25
4.5 ANALYSIS OF RESULTS	25
4.6 CONCLUSIONS	26
5 RESULTS	27
5.1 BASELINE RANKINGS	27
5.2 INITIAL RUN ACCURACY	28
5.3 POST-USER FEEDBACK RUN ACCURACY	29
5.4 IMPROVEMENT FROM INITIAL RUN TO POST-USER FEEDBACK RUN	30
5.5 ISSUES OF CONCERN	31
6 CONCLUSIONS AND FUTURE DIRECTIONS	33
6.1 QSEARCH	33
6.2 FUTURE RESEARCH	34
REFERENCES	37
APPENDIX 1: QUERY RELEVANCE CRITERIA	39
APPENDIX 2: RELEVANCE INDICES	41
APPENDIX 3: BASELINE RANKINGS	43
APPENDIX 4: INITIAL RUN RANKINGS	45
APPENDIX 5: POST-USER FEEDBACK RANKINGS	47
APPENDIX 6: IMPROVEMENT FROM INITIAL RUN TO POST-USER FEEDBACK RUN	49
ACADEMIC VITA	51

## List of Figures

FIGURE 2.1	SAMPLE SCHEMA HIERARCHY WITH SUMMARIZATION OF SELECTED TERMS	8
FIGURE 3.1	EXAMPLE SSM HIERARCHICAL STRUCTURE IN QSEARCH	17
FIGURE 5.1	CHANGE IN AVERAGE RANKING ERROR	30

## List of Equations

EQUATION 2.1	SEMANTIC DISTANCE METRIC	7
EQUATION 3.1	INITIAL RANK	16
EQUATION 3.2	CORRECTION FACTOR	18
EQUATION 3.3	RELEVANCE INDEX	20
EQUATION 5.1	BASELINE INDEX	27

# 1 Introduction

Given the vast amount of information available on the World Wide Web, the most difficult challenge facing users is locating the information they want. Search engines provide users with a way of querying a database of web information, but there are some limitations to their ability. One such limitation is their search space; only a small fraction of the information on the World Wide Web is contained in any one search engine's database. Another limitation is that search engines often operate by providing results that contain the *words* that comprise the user's query but not the *meaning* of those words. That is to say, they mainly perform a keyword search instead of a semantic search.

The goal of World Wide Web meta-search engines is to give users more relevant results by taking advantage of the combined search spaces of other search engines. However, it would be impractical for them to submit a user's query to every one of the many search engines available. Meta-search engines thus must choose a subset of available search engines; ideally, this subset would consist of the search engines with the best capabilities in the subject area of the user's query.

One way to provide subject area identification is to employ a semantic model called the Summary Schemas Model, which was originally developed for the field of multidatabases. The Qsearch algorithm developed at The Pennsylvania State University uses the Summary Schemas Model, along with search engine profiles and user feedback, to determine a set of search engines that would provide the most relevant results to a given user query. No prior evaluations of the effectiveness of this algorithm have taken place. The objectives of this research were:

- to determine whether the search engine rankings produced by Qsearch are more accurate than

random search engine rankings

- to determine whether Qsearch produces more accurate rankings after it is provided with user feedback than before – in other words, whether it is able to learn from users

Chapter 2 provides background information on multidatabase systems, the Summary Schemas Model, World Wide Web search engines, and motivations for using the Summary Schemas model in search engine selection. Chapter 3 introduces the Qsearch algorithm, an implementation of search engine selection using the Summary Schemas Model. Chapter 4 explains the methodology used to evaluate the accuracy of the Qsearch algorithm. Chapter 5 presents the results obtained from this evaluation. Chapter 6 contains conclusions derived from this research and suggests some directions for further study.

## 2 Background

The Summary Schemas Model (SSM) was originally proposed as a means to resolve global queries in a multidatabase platform. However, its application is not limited to the logical integration of heterogeneous databases. This section presents an overview of the SSM and a description of its possible use in the proper selection of the commercial Search Engine.

### 2.1 Multidatabase Systems

Many different organizations use databases to store and access large, organized bodies of information. As the use of databases increases and users need to access a number of distinct databases to obtain the information they want, it has become desirable to integrate these distinct databases. One way to achieve such integration is through a multidatabase. A multidatabase system is an interface between a user and a collection of local database management systems (DBMSs). Such a system allows a user to query a large collection of heterogeneous and distributed local databases without having particular knowledge about the local databases. The applications of such an interface can be expected to grow as merging businesses, collaborating researchers, and the general public find the need for consolidation of various sources of information [1].

One important characteristic of a multidatabase is that it provides full site autonomy for each local DBMS. Unlike a distributed database, a multidatabase accesses local functions through the local DBMS external user interface, rather than through internal DBMS functions. The local DBMS maintains full control over local data, processing, and participation in the global system [1].

A multidatabase differs from an interoperable system by providing full database function to global users. An interoperable system, the most loosely coupled information-sharing system, does not support full database functionality (including query processing) [1]. By contrast, a multidatabase provides global users with a front end to an agglomerated “virtual” database.

Two distinct approaches have been taken to multidatabase design: the global-schema approach and the multidatabase-language approach. In global-schema design, also called “view integration,” an integrated summary of the local schemas is created by collecting the local schemas and resolving the semantic and syntactic differences among them [1]. The primary advantage of global-schema design is user-friendly global access: the user interacts with the multidatabase in the same familiar way he or she interacts with a single database and need not know the details of the local DBMSs or how to resolve the differences among them. One disadvantage of global-schema design is the intensive human labor required in designing and maintaining the global schema, since the global database administrator must be thoroughly familiar with all the input schemas, user requirements, and local schema changes. A second disadvantage of global-schema design is that the global schema can be a very large data object, making its replication a problem for nodes with limited storage capacity [1].

A multidatabase language system shifts the responsibility for local DBMS integration from the global database administrator to the user. This system does not include an integrated summary of the local schemas. Instead, a common name space is defined, and the user is presented with language functions that specify data sources, transform source information into different representations, and control the result format. The advantages of a multidatabase language system are highly customizable user queries, less knowledge and development and maintenance time required of the global administrator, and minimal

storage requirements for global data at the local nodes. On the other hand, the user must have precise notions about the nature and location of the desired information and a willingness to undertake some programming to obtain this information [1].

## 2.2 The Summary Schemas Model

One of the key challenges in creating either type of multidatabase system is integrating local databases, in which information of varying consistency, completeness, and level of abstraction is stored using different naming conventions, formats, and data structures. Ideally, the multidatabase interface would undertake a user query by characterizing the semantic – rather than merely the syntactic – content of the query. Also, the integration of the local databases should not place undue burden on a global database administrator, who presumably is less familiar with the semantic content of the local databases than the local database administrators would be [1].

One solution to the problem of global integration of local heterogeneous databases is the Summary Schemas Model (SSM). Using online linguistic tools and a global hierarchy of summary schemas, the Summary Schemas Model identifies semantically similar entities in the local databases and automates the global integration process [2].

In order to identify linguistic relationships between terms in heterogeneous databases, the SSM uses a taxonomy appropriate to the range of user queries expected for the multidatabase. For instance, if a broad range of queries is expected, the SSM might use the taxonomy represented by Roget's thesaurus. The taxonomy itself is a collection of entries, each of which contains a term, a precise definition for the term, and semantic links to related terms. (In the example using Roget's thesaurus, the taxonomy would require

a supplementary source of definitions for terms.) At each node, the semantic links include a hypernym pointer, a list of hyponym pointers, and a list of synonym pointers [2].

The core of the SSM is its global metadata: the hierarchy of summary schemas. The physical network hierarchy generally determines the structure of this hierarchy, although other configurations that ensure good performance between parent and child nodes are also possible. The leaf nodes are database schemas for the individual local databases. Nodes higher in the hierarchy contain summary schemas, or collections of the hypernyms of their child nodes' terms. These higher-level nodes also contain copies of the "operational taxonomy," an abbreviated form of the full taxonomy that lacks definitions and substitutes identification numbers for textual terms [2].

Figure 2.1 presents a sample SSM hierarchy. At the lowest level are nodes containing local database schemas. One level higher are subhead nodes, which abstract the information contained in a collection of local databases. In Figure 2.1, the term 'Earnings' at node 4.A abstracts the ideas of 'Wage' and 'Salary,' which are represented as attributes in the local database schemas at nodes A and B, respectively. Each of the three higher levels – head, section, and class – in turn summarize the information contained in the immediately lower level. In the example, the head term 'Acquisition' is an abstract term encompassing 'Earnings' and 'Gain,' the section term 'Possessive Relations' summarizes 'Acquisition' and 'Payment,' and the class term 'Volition' summarizes 'Possessive Relations' (and presumably some other section-level terms).

After the summary schema's hierarchy has been created, all that is required for a local database to join the multidatabase system is for the local database administrator to map the terms in the local schema to entry-level terms in the taxonomy (using the definitions provided by the full taxonomy, if necessary). The

hypernym links in the hierarchy make addition and deletion of leaf nodes automatic [2].

As previously noted, one of the major disadvantages of regular multidatabase language systems is that users are expected to know the location and local-access terms for the data they seek. That is to say, users must submit queries containing *precise* data references. These queries are parsed at their origin nodes, which then request the necessary data from remote data sources and perform user-specified operations on the results.

Under the Summary Schemas Model, however, the user may submit *imprecise* queries – queries lacking location information and describing data in terms that the user deems semantically accurate, even if the terms do not exactly match the local-access terms for the desired data. When the query origin node recognizes a query marked as imprecise, it uses the summary schema structure to find and substitute semantically similar precise references for the imprecise references. The user can control how “similar” the substituted references should be by specifying a Semantic-Distance Metric (SDM) value. Each hypernym-hyponym relationship and synonym-synonym relationship is considered to be a semantic link; any two terms in the SSM taxonomy are associated with each other through some combination of semantic links. The Semantic-Distance Metric defines semantic distance as follows:

$$SDM = E(LC_i * LW_i)$$

(Equation 2.1)

where  $LC$  is the number of links between two terms  
 $LW$  is the weight (relative importance) of a link  
 $i$  represents the type of link (hypernym-hyponym or synonym-synonym)

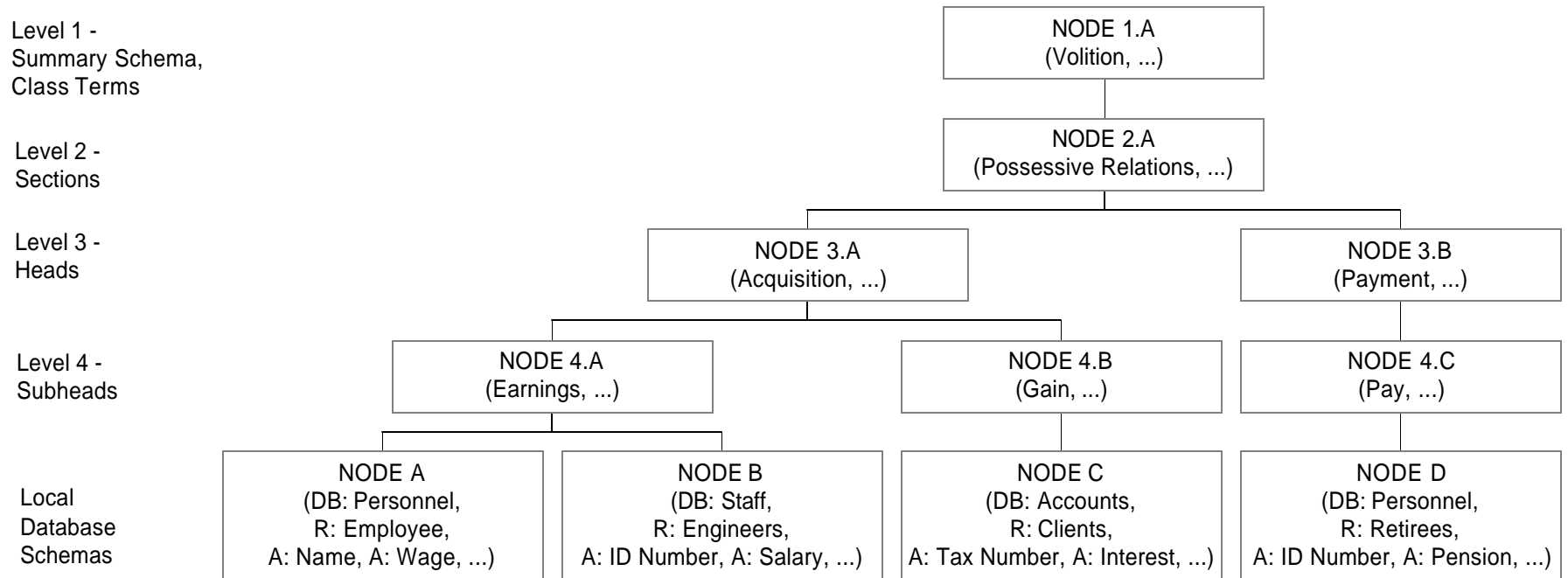


Figure 2.1: Sample schema hierarchy with summarization of selected terms

The SDM, then, is a weighted measure of the distance between two terms in the semantic hierarchy. A low value for the SDM would indicate that two terms are semantically similar; conversely, a high value would indicate that two terms are semantically disparate. The weight assigned to a given type of semantic relationship (represented in the equation above by  $LW_i$ ) varies inversely with the degree of semantic similarity it is judged to represent. For instance, if one decides that a word is generally closer in meaning to its synonym than to its hypernym or hyponym, one would assign a lower weight to the synonym-synonym links than to the hypernym-hyponym links [2].

The Summary Schemas Model permits imprecise query processing, summarizes information at each level of a hierarchy, utilizes a broad taxonomy for automatic semantic identification, and maps readily to popular computer network hierarchies. It thus fulfills the need for multidatabases to process queries based on their semantic content, maintain small global data structures, maximize the automation of multidatabase maintenance, and take advantage of existing network topology among the participating local databases [3].

### 2.3 World Wide Web Search Engines

The field of Internet services offers many potential applications of the Summary Schemas Model. In particular, both Internet users and multidatabase users want “timely and reliable access to heterogeneous and distributed data sources” [4].

The World Wide Web is a public, Internet-based distributed information system that allows users to navigate through hypertext documents. Introduced to the public in 1991, the World Wide Web spanned over twenty million active web servers in September 2000 [5]. Analysts forecast that the Internet, of which the World Wide Web is a major part, will have over 350 million users by the end of the year 2000 [6].

With the volume of content and number of new users on the World Wide Web increasing continually, the need for a way for users to find the information they seek is increasingly apparent. The answer to this need is a tool called a search engine. Search engines provide users with an interface for finding information on the World Wide Web.

Search engines can be grouped into three major categories, according to the methods they use to retrieve information [7]: software robots, directories, and meta-search engines.

### 2.3.1 Robot-based Search Engines

Search engines based on software robots “crawl” the World Wide Web, retrieving documents and entering them into a database that a user can query. Examples of robot-based search engines include Altavista[8], Google[9], and Excite[10]. Four basic components comprise robot-based search engines [11]: a spider, a summarizer, an indexer, and a broker. First, the spider retrieves Web documents, often finding new documents by following hyperlinks from documents that it has already discovered. The summarizer component then attempts to distill the document so that its content is retained and extraneous information (such as images and page formatting) is discarded. The indexer adds the summarized document to the search engine’s database, while the broker acts as an interface to allow users to query this database.

### 2.3.2 Directory-based Search Engines

Directories are another class of search engine. These search engines are hierarchical listings of topics and web sites that involve those topics. Yahoo[12], LookSmart[13], and the Open Directory

Project[14] are examples of directory-based search engines. Generally, directories require human administrators to add web site listings and maintain existing listings. In this case, people act as robots and indexers; directories do not have summarizers, since their listings indicate only the title of the web site or document, in addition to a link to it. Many, if not most, major web portals include both robot-based and directory-based search engines.

### 2.3.3 Meta-search Engines

Meta-search engines attempt to maximize their search space – the set of indexed web documents that they can access – by sending a user’s query to multiple robot-based search engines. Metacrawler[15], SavvySearch[16], and ProFusion[17] are meta-search engines. By delegating to other search engines the tasks of developing and storing databases of web content, meta-search engines expand their search space (a meta-engine’s search space is the union of the search spaces of the search engines it queries) while avoiding the need for massive amounts of storage to maintain their own web content databases. Meta-search engines do, however, require some specialized modules: a dispatch mechanism, an interface agent, and a display mechanism [4]. When a user enters a query, the dispatch mechanism determines what search engines in the meta-search engine’s knowledge base will be used for the query; this step is referred to as search engine selection. The interface agent then translates the query into the syntax used by these search engines, submits the query to each of them, and parses the results. Finally, the display mechanism performs post-processing on all the results that were received; this mechanism might re-rank the combined results or remove duplicate results before displaying them to the user.

## 2.4 Motivations for Using the SSM in Search Engine Selection

In October 2000, *Yahoo!* listed well over one hundred robot-based search engines for the World Wide Web [18]. With so many potential search engines to query, a meta-search engine must choose some subset of known search engines to query. Most meta-search engines offer the user a choice between a user-specified set of search engines (from the meta-search engine's knowledge base) or a meta-engine-specified set. If the choice is left to the meta-search engine, it could choose to query random search engines or to query the fastest search engines in its knowledge base. Ideally, though, a meta-search engine would query the set of search engines that it determines are likely to provide the most relevant results. The challenge of providing this capability is referred to as the Search Engine Selection Problem [4].

One way to address the Search Engine Selection Problem is to assume that a regular search engine tends to produce relevant results for queries within a certain set of knowledge domains. This assumption makes the Summary Schemas Model a natural choice of algorithm for choosing an appropriate set of search engines for a meta-search engine to use. In the same way that a multidatabase can use the SSM to identify semantically similar data references, a meta-search engine could use the SSM to identify semantically similar knowledge domains.

## 2.5 Conclusions

Multidatabases and the World Wide Web have a number of characteristics in common. Both are large, distributed information repositories. Both contain data in different representations, in terms of language, numeric format, completeness, terms used to represent a given concept, and other areas. Both

afford complete local site autonomy. Finally, both have many users who find it useful to be able to query the global system. The Summary Schemas Model was originally designed to resolve global queries within a multidatabase system by recognizing the semantic – as opposed to just the syntactic – content of a query. The research presented in this paper attempts to evaluate the Summary Schemas Model as a means of enhancing World Wide Web meta-search engines by allowing them to recognize the semantic content of the queries they receive.

### 3 Implementation: Qsearch

A computer program called Qsearch is under development at The Pennsylvania State University to implement Search Engine Selection using the Summary Schemas Model. It uses a hypothetical database node system and a Roget's Thesaurus taxonomy to predict the relevance of results returned by four robot-based search engines (AltaVista[8], Excite[10], Infoseek[19], and Yahoo[12]) based on the semantic content of queries. The three major elements of Qsearch's architecture are the user interface, the SSM hierarchy, and the meta-table [20].

#### 3.1 User Interface

The Qsearch user interface is a continuous cycle of user input and program calculation. First, Qsearch prompts the user to assemble the node structure of the SSM hierarchy. A set of ten files, each containing information describing the contents of a local node (specifically, the node's identity number, parent node, number of terms, and the terms themselves), is provided with Qsearch; the program assembles the node structure using this information. The user may choose to add any subset of these nodes to the global hierarchy. (The terms contained in these nodes do not constitute the complete contents of Roget's Thesaurus, but a user may create his or her own node files and specify them in an execution of Qsearch.) Next, the user enters a query, as specified by a query string, a maximum Semantic Distance Metric, and a starting node. The user query is treated as an imprecise query; Qsearch resolves this query by beginning at the specified starting node and searching the name space represented by the summary schemas hierarchy, up to the specified maximum Semantic Distance Metric. Qsearch then calculates a 'relevance

index' for each of the four search engines in its knowledge base. This relevance index is a real number between zero and three that reflects Qsearch's prediction of the relevance of results returned by that search engine based on the semantic content of the query. The user then manually submits the query to the four search engines and provides Qsearch with feedback about the results. Specifically, for each search engine, Qsearch prompts the user for the number of links followed, and for each followed link, the search engine's ranking, the user's rating, and the web page title. Qsearch incorporates this feedback into its knowledge base and prompts the user for the next query.

## 3.2 SSM Hierarchical Structure

Qsearch uses its Summary Schemas Model hierarchy to produce an initial rank for each search engine for a given query. (The use of the qualifier "initial" here reflects the notion that Qsearch initially has limited information about the performance of each search engine in its knowledge base. This information is represented in the SSM Hierarchical Structure and is augmented with user feedback stored in meta-tables as described in Section 3.3.) Qsearch's hierarchy is only slightly modified from the SSM structure designed for multidatabase systems. A ranked list of search engines is maintained for each term in the top three levels of the SSM hierarchy. These static rankings were pre-determined by the program developer and reflect initial assessments of the ability of these search engines to return relevant results for the associated terms. In Figure 3.1, rank is represented by list order; the highest-ranked engine is first in the list, followed by the second-ranked engine, and so on.

For each term at a leaf node, there is at least one path leading from a top-level node to that term. Along any given path, the search engine rankings for the terms at the top three levels may differ. The

example in Figure 3.1 shows the conceptual content of part of Qsearch’s SSM Hierarchy. Here, the path to leaf-node term “consensus” from the top level is “intellect,” “results of reasoning,” “assent,” “unanimity,” “consensus.” Qsearch combines these rankings so that each level’s contribution to the overall ranking is inversely proportional to the number of hyponyms of that level’s term, based on the assumption that a term with fewer hyponyms is semantically closer to its hyponyms than a term with many hyponyms. Qsearch uses the following equations (note that a higher ‘rank’ indicates a search engine that returns more relevant results) to calculate the initial rank of each search engine:

$$initial\_rank(engineA) = r_{1A}/n_1n_2 + r_{2A}/n_2 + r_{3A}$$

$$initial\_rank(engineB) = r_{1B}/n_1n_2 + r_{2B}/n_2 + r_{3B}$$

$$initial\_rank(engineC) = r_{1C}/n_1n_2 + r_{2C}/n_2 + r_{3C}$$

$$initial\_rank(engineD) = r_{1D}/n_1n_2 + r_{2D}/n_2 + r_{3D}$$

(Equation 3.1)

where  $n_i$  is the number of hyponyms for the node term on level  $i$

$r_{YX}$  is the value associated with a ranking of search engine X on level Y of the SSM hierarchy:

$r_{YX} = 1.00$	if engine X ranked #1 on level Y
$r_{YX} = 0.75$	if engine X ranked #2 on level Y
$r_{YX} = 0.50$	if engine X ranked #3 on level Y
$r_{YX} = 0.25$	if engine X ranked #4 on level Y

For the example in Figure 3.1, Qsearch would calculate the initial rank for each search engine as follows (assuming that Figure 3.1 represents a complete taxonomy):

$$initial\_rank(Altavista) = (0.25)/3*5 + (0.50)/5 + (0.75) = 0.8670$$

$$initial\_rank(Excite) = (0.75)/3*5 + (0.25)/5 + (0.25) = 0.3500$$

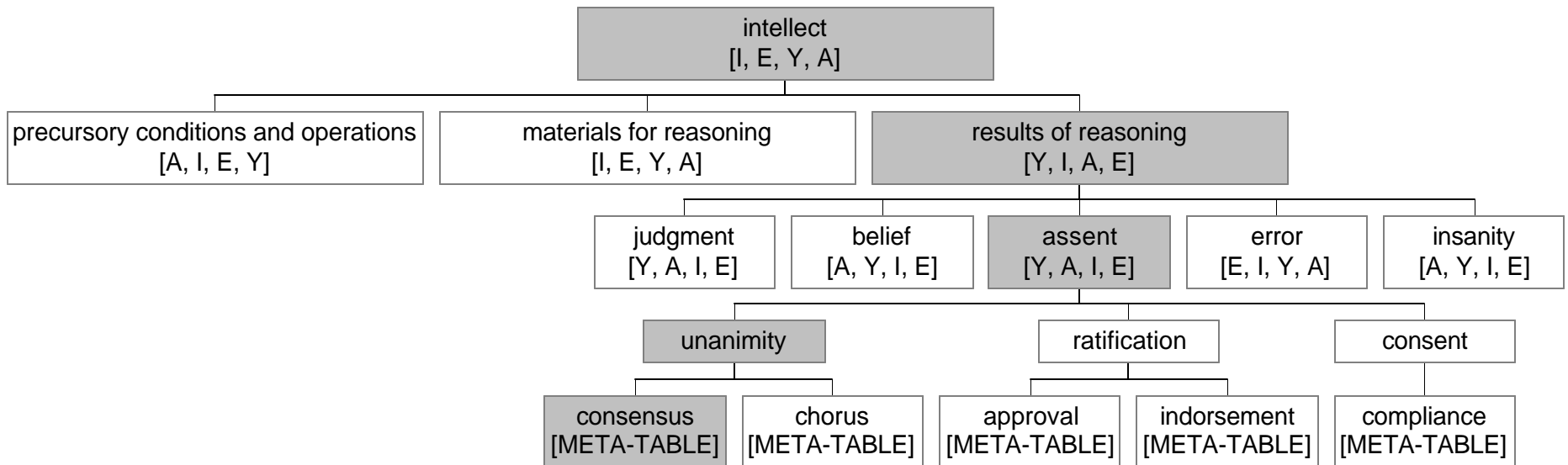


Figure 3.1: Example SSM Hierarchical Structure in Qsearch (A stands for Altavista, E stands for Excite, I stands for Infoseek, Y stands for Yahoo)

$$initial\_rank(Infoseek) = (1.00)/3*5 + (0.75)/5 + (0.50) = 0.7167$$

$$initial\_rank(Yahoo) = (0.50)/3*5 + (1.00)/5 + (1.00) = 1.2333$$

These calculations show that, given the SSM Hierarchy represented by Figure 3.1 and a query consisting of the term “consensus,” Qsearch would recommend the search engine Yahoo as likely to produce the most relevant results in that query’s semantic realm.

### 3.3 Meta-table

Qsearch dynamically updates its knowledge of search engine performance through user feedback, which is stored in the meta-table. A meta-table is maintained in memory for each leaf node in the SSM structure. This meta-table is a three-dimensional  $t \times e \times p$  matrix, where  $t$  is the number of terms at the leaf node,  $e$  is the number of search engines in Qsearch’s knowledge base (currently four), and  $p$  is the number of paths to the leaf node. Each meta-table entry contains a dynamically-updated correction factor to be added to the initial rank (determined by the SSM Hierarchical Structure with the Equation 3.1) to produce the final ‘relevance index’ presented to the user. This correction factor arises from previous user feedback on that query term, including such information as the number of followed links and an active user ranking, the search engine’s ranking, and the title terms for each followed link. The following equations determine the correction factor (CF) for a term:

$$CF = \frac{-1}{n}, k = 0 \quad CF = \frac{\sum_{i=1}^k e_i u_i S_i}{k}, k > 0$$

(Equation 3.2)

where  $k$  is the number of links followed  
 $n$  is the number of terms in the query  
 $e_i$  is a normalized engine ranking on a scale of 0 to 1 corresponding to the rank of link  $i$  on the search engine's list from 10 to 1  
 $u_i$  is the Qsearch user's ranking, from 0 to 1, of the relevance of link  $i$   
 $s_i$  is number of terms within the specified maximum Semantic Distance Metric value of the query term that appear in the title of link  $i$

The critical terms in Equation 3.2 are  $e_i$ ,  $u_i$ , and  $s_i$ . The purpose of  $e_i$ , the search engine's ranking of result  $i$ , is to give a higher weight to terms that appear higher on the search engine's result page. All four search engines included in this research present their results in an order based on a relevance index calculated by that search engine. The inclusion of this term " $e_i$ " reflects the idea that a search engine that produces relevant results must not only return relevant results but also accurately assess their relative relevance, since neither a user nor a meta-search engine would find it practical to sift through thousands (or even just dozens) of results to find the sought-after information. The term  $u_i$  reflects the importance of the user's own assessment of the relevance of each result. Finally, the term  $s_i$  is included to give greater weight to results whose titles include the query term or semantically similar terms.

If a Qsearch user provides feedback on a term that already has a correction factor, the newly-calculated correction factor replaces the old one. If Qsearch is presented with a query term for which it has no previous user feedback (that is, the meta-table entry for that term is empty), the correction factor is zero for that term.

The 'relevance index' for each search engine, which is calculated by Qsearch and presented to the user as a prediction of the degree of relevance of that search engine's results for the given query, is a sum of the initial rank and the correction factors:

$$relevance\_index(engine, terms) = initial\_rank(engine, terms) + \sum_{terms} CF(engine, term)$$

(Equation 3.3)

A high relevance index indicates that Qsearch predicts a high degree of relevance for that search engine's results, while a low relevance index indicates the prediction of a lower degree of relevance.

### 3.4 Conclusions

With a semantic hierarchy for initial ranks and a dynamically updated meta-table for user feedback, Qsearch uses the Summary Schemas Model to produce a quantitative valuation of the ability of each of a set of search engines to produce relevant results for a particular query. A commercial meta-search engine could use Qsearch to solve the Search Engine Selection Problem with the addition of a rule for determining to which search engines a query is actually submitted. Examples of such rules might include "submit query to the top  $n$  search engines," "submit query to all search engines with a relevance index above  $x$ ," and "submit query to the top  $r$  search engines with average response times above  $s$  seconds."

## 4 Evaluation Methods

It would be interesting to investigate whether an Internet meta-search engine that uses the Summary Schemas Model in choosing which search engines to query could produce more relevant results than a meta-search engine that does not use SSM. Naturally, the implementation details of the specific meta-search engine under scrutiny – how many and which search engines are included in its knowledge base, what policies it uses to choose the search engines to which a query is actually submitted (given some form of ‘assessment’ based on the SSM), and how it combines the results from these search engines – would affect such a judgment. The research conducted in this study sought to evaluate the effectiveness of the core SSM-based assessment, as opposed to the effectiveness of a full meta-search engine with its own particular implementation policies. Because the computer program implementing Qsearch was not complete at the time this research was conducted, this methodology in some parts involved a ‘pencil-and-paper’ version of the Qsearch algorithm instead of the automated, computerized version. Differences between the two versions and their anticipated effects on the results are noted. This evaluation used the Qsearch algorithm to generate SSM-based assessments of relevance for four popular Internet search engines and a broad range of user queries. This section describes how the search engines and queries were chosen, how Qsearch was executed, and how the results were analyzed to judge the effectiveness of Qsearch as a solution to the Search Engine Selection Problem.

### 4.1 Search Engine and Query Choices

The first step was the choice of search engines and queries. Altavista[8], Excite[10], Infoseek[19]

(now Go.com[21]), and Yahoo[12] were chosen as the search engines because they constituted Qsearch's pre-existing knowledge base. These four search engines are among the most popular search engines used by meta-search engines and direct users alike; a February 2000 report at the web site Web Space Station ranks Yahoo, Go.com, Excite, and Altavista as the first, second, third, and fifth most popular Internet search engines, respectively, for queries in the English language originating in the United States [22]. A set of fifty query terms was then randomly chosen from the lowest level of Qsearch's taxonomy (see Appendix 1 for a list of these terms). Each query consisted of exactly one term, although some terms consisted of multiple words; for example, "yellow pages" was considered to be a single term.

## 4.2 Initial Run

After the search engines and queries were chosen, the initial run of the Qsearch algorithm was executed. The purpose of this execution was to determine the relevance indices that Qsearch would calculate based only on its pre-existing knowledge base – that is, based on the rankings of the four search engines that it stores as persistent data for each term in the top three levels of the Summary Schemas Model hierarchy. The portion of the Qsearch computer program that calculates these initial relevance indices (by using Equation 3.1) was operational when this research was conducted, and so this step was automated. At each execution of Qsearch, all ten available local schemas (as described in Section 3.1) were added to the global schema before queries were processed. For each query, the maximum allowable semantic distance metric was set to five, and the starting node was set to zero. In the initial run, then, Qsearch was initialized, each of the fifty queries was processed according to the equations in Section 3, and the resultant initial relevance index was recorded.

### 4.3 User Feedback

Once Qsearch produced an initial relevance index for each pairing of search engine and query, that query was submitted to the search engine to determine the actual relevance of the results it produced. No quotation marks or other special symbols were included in the submitted queries. The links for each of the top ten results were followed, and the title of the resultant web page and the user's rating of the relevance of this result were recorded and input to the Qsearch algorithm as feedback.

The user's rating was a real number between zero and one that reflected a subjective judgment of the usefulness of the result page (or any page no more than one link away from it) to a user seeking general information about the topic described by the query. Appendix 1 lists loose guidelines for the types of information judged to be relevant for each query. In general, a page was judged to be irrelevant if it or its web server could not be found or accessed (also known as a 'dead link'); if accessing relevant information would have required downloading a document in a non-standard web format (i.e., a non-HTML, non-image document) or making a purchase; if the document was not in the English language; if the document used the term only as a proper noun not related to the denotation or connotation of the term; or if the document was pornographic. Duplicate results – that is, the same web page or pages from the same web site appearing more than once in a search engine's list of top ten results – were each assigned the full user rating value ascribed to that result.

In addition to the user's and search engine's ratings of each result, the 'semantic match' between the title of the result page and the query term is considered in the Qsearch algorithm's calculations. The concept of 'semantic match' is represented in Equation 3.2 by the term  $s_i$ , the number of terms appearing in the result page title that are within the specified maximum Semantic Distance Metric of the query term

for result  $i$ .

The Qsearch computer program's string comparison capabilities are limited; it is case-sensitive and substring-insensitive, meaning, for example, that it does not recognize the semantic similarity between 'canine' and 'Canine' or between 'canine' and 'canines.' In order to compensate for this limitation, the result page titles must be input to Qsearch in a form that would allow the program to recognize semantic similarities where they clearly existed. For instance, upper-case words must be converted to lower-case, and words containing the query term as a substring must be partitioned so that the query term appeared as a separate word.

The user feedback step was performed without the aid of the Qsearch computer program, since the meta-table component of the program was not operational at the time of this research. Correction factors were thus calculated by hand, introducing some complication into the determination in Equation 3.2 of the value of  $s_i$ , the number of terms in the result page title that are within the specified maximum Semantic Distance Metric of the query term for result  $i$ . Because of the difficulty of reconstructing the Qsearch program's entire Summary Schemas Model hierarchy, a subjective estimate of semantically similar terms contained in each result page title was made. For example, it was estimated that the term 'dog' is probably not farther than a Semantic Distance Metric of five from the term 'canine,' and so a web page title containing both terms was assigned a  $s_i$  value of two. If these approximations affected the results, it was likely that they provided somewhat higher  $s_i$  values than the computer program would have provided, since the program's data set included only a subset of the terms in Roget's Thesaurus, while human judgment allows a more flexible, 'common sense' approach to semantic similarity. Still, care was taken to assign  $s_i$  values that would reflect the limitations inherent in the computer program.

#### 4.4 Post-User Feedback Run

After the user feedback was provided to the Qsearch algorithm, the algorithm was run through a post-user feedback run on the same fifty queries that were processed in the initial run. The purpose of this post-user feedback run was to determine the relevance indices that Qsearch would calculate once it was provided with user feedback on specific queries, in addition to the pre-existing information about search engine performance for general subject areas. Equation 3.3 was calculated in this step; that is, the correction factors calculated in the user feedback step using Equation 3.2 were added to the relevance indices calculated in the initial run using Equation 3.1. As a result, new relevance indices were produced that reflected user feedback. The summation was performed by hand instead of using the Qsearch computer program, but this process is not expected to have introduced any discrepancy between the reported results and the results that would be expected from the automated program. Appendix 2 lists the initial run and post-user feedback run relevance indices generated by the Qsearch algorithm for each pairing of query and search engine.

#### 4.5 Analysis of Results

The relevance indices collected in the initial run and post-user feedback run steps as well as the user rating of the relevance of each result were then used in a statistical analysis to answer a number of questions about Qsearch's performance. Qsearch's ability to accurately predict the best search engine(s) for a query was assessed by comparing the rankings evident in the user ratings to the rankings evident in the relevance indices for the initial and post-user feedback executions of the program. Moreover,

Qsearch's ability to learn – that is, to dynamically correct its relevance indices based on user feedback – was assessed by analyzing the change in prediction accuracy from the initial run to the post-user feedback run.

## 4.6 Conclusions

The methodology described in this section was designed to produce an accurate evaluation of the SSM-based algorithm Qsearch as a solution to the Search Engine Selection Problem. By analyzing the precision of Qsearch's search engine rankings (both before and after user feedback) against a baseline of user rankings, which are presumed to be the most accurate reflection of result relevance, one can draw conclusions about Qsearch's ability to predict suitable engine(s) for a given query. Further, by analyzing the variation in accuracy between Qsearch's pre-feedback predictions and its post-feedback predictions, one can draw conclusions about Qsearch's ability to learn from the user feedback it receives. The conclusions based on Qsearch's performance are particularly meaningful to meta-search engine designers, who continually seek better ways to choose from the myriad search engines available. This methodology was developed to test the viability of making such a choice based on the semantic content of a query.

## 5 Results

In this section, the accuracy of Qsearch’s search engine rankings is evaluated by comparing the initial run rankings and post-user feedback rankings to a set of baseline rankings. First, the calculation of the baseline rankings is explained. Following this explanation are summaries of the accuracy of the initial run rankings, the accuracy of the post-user feedback rankings, and a comparison of the pre- and post-feedback accuracies. Thorough supporting data is provided in the Appendices.

### 5.1 Baseline Rankings

The idea of the baseline rankings is to give an assessment of how an average user would rank the search engines given the time to follow and evaluate each of the top ten results provided by each engine.

For each query, Appendix 3 lists the baseline search engine rankings, those rankings considered to be ‘accurate’ and against which the rankings provided by Qsearch both before and after user feedback are compared. These rankings were calculated based the following equation:

$$baseline\_index(engine, term) = \frac{\sum_{i=1}^{10} e_i u_i}{10}$$

(Equation 5.1)

In this equation, accuracy is represented by a combination of the placement of a result within the

search engine's ranked list of results ( $e_i$ ) and the user's rating of the relevance of that result ( $u_i$ ), for the top ten results. These terms have the same meanings and domains as their counterparts in Equation 3.2. The search engine with the highest baseline index, as calculated using Equation 5.1, was assigned a rank of one, the search engine with the second-highest baseline index was assigned a rank of two, and so on.

Statistical analysis was used to determine the average error (with respect to these baseline rankings) of random sets of rankings. The purpose of this calculation was to ascertain whether the rankings produced by Qsearch are significantly more accurate than random rankings would be. One thousand sets of random rankings were generated, and the average error of each of these rankings was calculated. At a 95% confidence level, the average error of random rankings was found to be between 1.125 and 1.395.

## 5.2 Initial Run Accuracy

Appendix 2 lists the relevance indices produced in the initial run of the Qsearch algorithm. Appendix 4 lists the rankings into which those relevance indices translated (with a rank of one indicating the search engine with the highest predicted relevance, a rank of two for the search engine with the second-highest predicted relevance, et cetera). The table in Appendix 4 also shows the 'error' in these rankings – that is, for each search engine, the difference between its baseline ranking (listed in Appendix 3) and its initial run ranking. The average error for each query is displayed in the rightmost column of this table; this average is calculated by dividing the sum of the absolute values of the individual errors by the number of search engines (four).

The average error for the initial run rankings over all search engines and all queries was 1.27. In other words, for any choice of query and search engine (out of those included in this study), one can expect the search engine ranking provided by an initial run of the Qsearch algorithm to differ from the ranking that would be calculated by his or her own judgment by an average of 1.27 ranks. This error is not significantly different from the error of randomly-generated rankings. Statistical analysis shows that a randomly-generated set of rankings has a 57% chance of having a lower error level than the initial run rankings and a 43% chance of having a higher error level.

### 5.3 Post-User Feedback Run Accuracy

Appendix 2 lists the relevance indices produced in the post-user feedback run of the Qsearch algorithm. Appendix 5 lists the rankings into which those relevance indices translated (with a rank of one indicating the search engine with the highest predicted relevance, a rank of two for the search engine with the second-highest predicted relevance, et cetera). The table in Appendix 5 also shows the error in these rankings, as explained in Section 5.2.

The average error for the post-user feedback run rankings over all search engines and all queries was 1.06. In other words, for a random choice of query and search engine (out of those included in this study), one can expect the search engine ranking provided by a post-user feedback run of the Qsearch algorithm to differ from the ranking that would be calculated by his or her own judgment by an average of 1.06 ranks. This error is significantly lower than the error of randomly-generated rankings. Statistical analysis shows that a randomly-generated set of rankings has a 4% chance of having a lower error level

than the initial run rankings and a 96% chance of having a higher error level.

#### 5.4 Improvement from Initial Run to Post-User Feedback Run

Appendix 6 lists the average error of the rankings for each query in the initial run and in the post-user feedback run. In addition, this table shows the amount of decrease in the average ranking error for each query, as well as the amount of change (the absolute value of the amount of decrease).

As illustrated in Appendix 6, the average error of the rankings for each query decreased by an average of 0.21 ranks. It is unclear whether this change would be noticed by a normal user. The tendency of the error to decrease (or at least not increase) from the initial run to the post-user feedback run appears to be strong, however, as evidenced by the fact that the average error decreased for 21 of the queries (42%), stayed the same for 26 of the queries (52%), and increased for only 3 of the queries (6%) (see Figure 5.1).

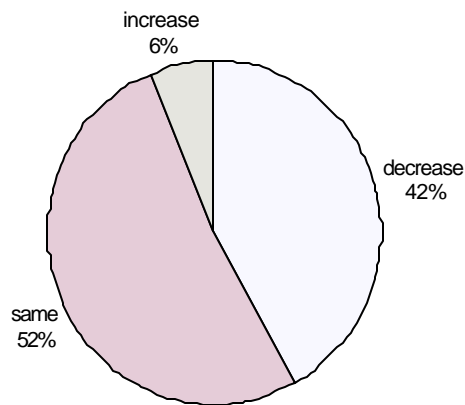


Figure 5.1: Change in Average Ranking Error

## 5.5 Issues of Concern

Several sources of error could have affected the results described in the previous three sections. First, the criteria for relevance for each query (listed in Appendix 1) were fairly narrow relative to the broad scope of the terms in general. For this reason, they may have prevented the ‘true’ relevance of a search engine’s results from being shown. A one-term query, such as those included in this study, is not nearly as descriptive of what a user hopes to find as is a multi-term query; what one user considers a relevant result for a query on the term ‘photograph’ may be completely irrelevant to another user submitting a query on the same term.

Second, the Qsearch algorithm used in this study included a knowledge base of only four search engines. The addition of more search engines to the knowledge base would increase the potential for more relevant results, particularly if the additional search engines are ones that specialize in certain subject areas. With only four general-knowledge search engines used, queries tended to produce similar results on different search engines, with the consequence that the ability of some search engines to produce markedly superior results in certain subject areas could have been underestimated.

Third, the ‘semantic match’ value that contributes to a term’s correction factor (see Section 3.3) is complicated to calculate accurately. This value equals the number of terms in a result web page’s title that are within the specified maximum Semantic Distance Metric (SDM) of the query term(s) in Qsearch’s Summary Schemas Model hierarchy. It is not difficult for a computer program to traverse the hierarchy to determine whether a word is within the specified SDM of a query term. Rather, the difficulty lies in the program’s ability to recognize a given word in a web page title, where it may be in upper-, lower-, or mixed-case; in any one of a number of forms (plural, past tense, and so on); concatenated with another

word; or part of an acronym or abbreviation. This limitation discriminates against search engines which do recognize these other forms of words by, for example, providing results including the title 'USA' for a query on 'United States of America.'

A fourth possible source of error is that Qsearch treats all the results provided by a search engine for a given query as distinct; it does not recognize duplicate results. Even if a certain result is relevant to a user's query, that same result appearing again in a search's engine's top ten results does not provide any additional information value to the user. Depending on whether the duplicate results are duplicates of relevant or irrelevant original results, this behavior could discriminate in favor of or against search engines that tend to provide duplicate results.

## 6 Conclusions and Future Directions

The Summary Schemas Model (SSM) provides global imprecise query resolution for multidatabases by identifying the semantic content of queries. Because of this semantic identification ability, SSM could potentially be used by World Wide Web meta-search engines to choose a set of search engines for query submission that would provide the most relevant results, based on the semantic content of the query. Qsearch is an algorithm that uses SSM and user feedback to provide semantically-based, dynamically-updated rankings of a set of Web search engines. An evaluation of Qsearch's ability to provide accurate rankings and to learn from user feedback was conducted, with important ramifications for meta-search engine designers.

### 6.1 Qsearch

The evaluation of Qsearch found that this algorithm performed poorly at providing accurate search engine rankings without user feedback, and its ability to learn is somewhat limited. The accuracy of the search engine rankings produced by Qsearch without user feedback was consistent with the accuracy of randomly-generated rankings. Thus, the Qsearch algorithm without user feedback does not appear to be a useful predictor of search engine relevance. Also, for a majority of the queries included in this study (52%), the average error of Qsearch's rankings (relative to a user's) was the same after user feedback as before.

However, Qsearch does show some promise as a solution to the Search Engine Selection Problem. The accuracy of the search engine rankings produced by Qsearch with user feedback was significantly

higher than the accuracy of randomly-generated rankings. In addition, the average error of Qsearch's rankings decreased by about one-fifth of a rank per search engine rank when user feedback was provided to the algorithm. In addition, the average error decreased for 42% of the queries after feedback was provided, while it increased for only 6% of the queries.

In summary, while Qsearch's search engine rankings prior to user feedback are not accurate, its accuracy does improve when the algorithm is provided with user feedback. Qsearch thus has the potential, given enough user feedback, to aid meta-search engines in choosing the best set of search engines for any given query.

## 6.2 Future Research

A number of areas have been identified for further work on Qsearch. The three main components of such future research are research into Qsearch's premises, refinement of the Qsearch algorithm itself, and completion and improvement of the computer program implementation of Qsearch.

The Qsearch algorithm is based on several premises which are supported more by general belief than by thorough research. One of these premises is the idea that some search engines produce more relevant results than others in various subject areas defined by the higher-level terms in Roget's Thesaurus. Future work could involve a statistical comparison of search engines based on query terms from a number of subject groupings. Another premise is that Roget's Thesaurus is an appropriate taxonomy to describe the content of the World Wide Web. On the web, many common words, like 'chat' and 'map,' have been re-appropriated to take on new meanings which are not reflected in traditional, pre-Internet taxonomies. Further work could be done to develop a web-appropriate taxonomy or to investigate candidates for such

a taxonomy, possibly including current directory-based search engines. A third unexamined premise is Qsearch's expectation of user behavior. Specifically, Qsearch expects that most users only look at the first ten results returned by a search engine and that the attention they pay to each result, from first to tenth, decreases geometrically. Future research might review literature on search engine user interfaces and usage patterns to obtain a realistic model of how people navigate through search engine results.

A second area of potential work is refinement of the Qsearch algorithm itself. Qsearch currently does not specify how to calculate the initial rank (see Equation 3.1) for a multi-term query. Further research could investigate different ways to perform this calculation. Also, each new entry in its meta-table replaces the previous entry, with the effect that, for example, an entry based on a user examining ten results could be replaced by an entry from a user examining only one. More research remains to be done on ways that the correction factors in these meta-table entries could be accumulated to reflect multiple instances of user feedback for a given term and search engine. In addition, the usefulness of the 'semantic match' term for the web page title in the calculation of these correction factors should be researched, as it is not obvious that this term provides useful or accurate information about the relevance of a search engine result. In the course of this evaluation, it was often the case that the title of a relevant result page did not contain terms that Qsearch could identify as semantically similar to the search term. It also often happened that the titles of irrelevant result pages did contain the search term, such as when a query on the term 'frog' produced result pages titled 'Radioactive Frog Web Designs,' 'Squished Frog Productions,' and 'Purple Frog Software.' Future research could examine the utility of this 'semantic match' value and the correlation between relevant results and results with high semantic match values.

The computer program implementing Qsearch is not yet complete, but a number of issues related

to it remain to be investigated. One issue is that of user feedback: Qsearch's architecture calls for a tedious manual process of assessing search engine results and typing in their titles in order to calculate the correction factor for a term. Providing a convenient user interface for providing user ratings and automating the input of web page titles would greatly improve the usability of this program. Some additional efforts that could expand the program's scope and raise its accuracy include adding more search engines to its knowledge base, providing it with the full Roget's Thesaurus taxonomy, and enhancing its string matching capabilities for recognizing semantic matches.

## References

- [1] Hurson, A.R., and Bright, M.W. "Multidatabase Systems: An Advanced Concept in Handling Distributed Data," *Advances in Computers*, Vol. 32, 1991, pp. 149-200.
- [2] Bright, M.W., Hurson, A.R., and Pakzad, S.H. "Automated Resolution of Semantic Heterogeneity in Multidatabases," *ACM Transactions on Database Systems*, Vol. 19, No. 2, 1994, pp. 212-253.
- [3] Dash, K.I. and Hurson, A.R. "The Semantic Matrix Model (SMM): A Knowledge Based Solution to Semantic Homogeneity in Multidatabases," *International Conference on Information and Knowledge Management*, 1995, pp. 122-128.
- [4] Hurson, A.R., and Smith, L. "Meta-Search Engines." Department of Computer Science and Engineering, The Pennsylvania State University. Unpublished document.
- [5] "Netcraft Web Server Survey." Sep. 2000. Netcraft. 29 Oct. 2000  
<http://www.netcraft.co.uk/survey/>
- [6] "Cyberatlas: Geographics - The World's Online Populations." 18 Sep. 2000. Cyberatlas. 19 Sep. 2000  
[http://cyberatlas.internet.com/big\\_picture/geographics/article/0,1323,5911\\_151151,00.html](http://cyberatlas.internet.com/big_picture/geographics/article/0,1323,5911_151151,00.html)
- [7] Hurson, A.R., and Smith, L. "Commercial Search Engines." Department of Computer Science and Engineering, The Pennsylvania State University. Unpublished document.
- [8] *Altavista*. <http://www.altavista.com>
- [9] *Google*. <http://www.google.com>
- [10] *Excite*. <http://www.excite.com>
- [11] Wendling, Mike. "WWW Search Engines." *SLAC Database Forum & SWUG*. 26 Jul. 1996. 17 Sep. 2000. <http://www.slac.stanford.edu/~wendling/sld001.htm>
- [12] *Yahoo*. <http://www.yahoo.com>
- [13] *LookSmart*. <http://www.looksmart.com>
- [14] *Open Directory Project*. <http://www.dmoz.org>

- [15] *Metacrawler*. <http://www.metacrawler.com>
- [16] *SavvySearch*. <http://www.savvysearch.com>
- [17] *ProFusion*. <http://www.profusion.com>
- [18] “Yahoo! Computers and Internet > Internet > World Wide Web > Searching the Web > Search Engines.” 29 Oct. 2000.  
[http://dir.yahoo.com/Computers\\_and\\_Internet/Internet/World\\_Wide\\_Web/Searching\\_the\\_Web/Search\\_Engines/](http://dir.yahoo.com/Computers_and_Internet/Internet/World_Wide_Web/Searching_the_Web/Search_Engines/)
- [19] *Infoseek*. <http://www.infoseek.com>
- [20] Hurson, A.R., and Smith, L. “Qsearch: A Proposed Solution to the Search Engine Selection Problem.” Department of Computer Science and Engineering, The Pennsylvania State University. Unpublished document.
- [21] *Go.com*. <http://www.go.com>
- [22] “Search Engine Popularity Reporter.” 1 Feb. 2000. Web Space Station. 13 Nov. 2000.  
[http://www.web-space-station.com/cgi\\_bin/se-reporter/se-reporter.cgi?sort=0&country=USA&language=English&x=23&y=15&sort=2#report](http://www.web-space-station.com/cgi_bin/se-reporter/se-reporter.cgi?sort=0&country=USA&language=English&x=23&y=15&sort=2#report)

## Appendix One: Query Relevance Criteria

<b>Query</b>	<b>Relevance Criteria</b>
canine	physical characteristics; species
chatter	definition; examples
jest	definition; examples
frog	physical characteristics; photographs; types
united_states_of_america	map; history; almanac-type facts; regions; states
laughingstock	definition; examples
map	images; links to atlases; general description; parts and symbols
lyric	poetry; music lyrics
dictionary	definition; links to dictionaries; titles of dictionaries
weather	climate; forecasts; precipitation
harmonics	vibration; frequency; nodal points; musical instruments
wallpaper	definition; ways to apply; popular patterns; images
lodgings	hotel chains
salutation	definition; examples
love	definition; expressions of love
interval	length of time or distance
friendship	definition; examples of friends
position	job opening/description; career level
password	guidelines for choosing; how passwords are kept secret
horoscope	examples
motion_picture	movie titles; production; history; technical details
photograph	cameras; photography techniques
baptism	description; which religions practice it; who is baptized
automobile	makes; models; operation
quote	lists of quotes; how to include a quote in a paper
maid	what services a domestic hired maid provides

remedy	medicine; treatments; corrections
vocal_music	genres; forms; vocalists; history; examples
yellow_pages	description; example or link to example
timeliness	definition; example
buyer	advice; buyer's guide
caricature	history; definition; graphic example
marijuana	chemical description; laws; history; usage
reproof	definition; example
matter	states; properties; relationship to energy; building blocks of matter
population	statistics; methods of measurement
Muses	names and descriptions
voyage	description; account
marketplace	location; what is sold there
task	example
audience	importance; reaction to performance
paper	types; production; material composition
search	Internet search engines; methods; account
parts_of_flower	description; diagram
airline	commercial airline companies
preteen	sites for preteens; advice for parents
plundering	historical account
wager	gambling; advice (how much to wager)
infant	how to care for; physical and mental development
ballot	types; means of secrecy; policies

## Appendix Two: Relevance Indices

Query	Initial Run				Post-User Feedback Run			
	Altavista	Excite	Infoseek	Yahoo	Altavista	Excite	Infoseek	Yahoo
canine	1.02	0.54	0.78	0.52	1.1	0.63	0.84	0.59
chatter	0.27	0.79	1.03	0.52	0.27	0.79	1.03	0.52
jest	1.02	0.54	0.78	0.52	1.04	0.62	0.79	0.74
frog	1.02	0.54	0.78	0.52	1.17	0.99	1.15	0.87
united_states_of_america	0.77	1.04	0.78	1.02	1.14	1.33	1.12	1.51
laughingstock	0.76	1.01	0.25	0.51	0.76	1.12	0.29	0.69
map	0.27	0.79	1.03	0.52	0.3	1.29	1.25	0.59
lyric	0.27	0.79	1.03	0.52	0.42	1.35	1.4	0.61
dictionary	0.27	0.79	1.03	0.52	0.8	1.3	1.42	1.02
weather	1.02	0.54	0.78	0.52	1.53	1.09	1.3	1.07
harmonics	0.02	0.04	0.03	0.02	0.13	0.27	0.25	0.15
wallpaper	0.77	1.04	0.78	1.02	0.82	1.04	0.96	1.02
lodgings	0.77	1.04	0.78	1.02	1.31	1.6	1	1.81
salutation	1.02	0.54	0.78	0.52	1.05	0.56	0.81	0.66
love	1.02	0.54	0.78	0.52	1.19	1.19	0.87	0.86
interval	0.77	1.04	0.78	1.02	0.96	1.24	0.91	1.46
friendship	1.02	0.54	0.78	0.52	1.35	0.73	1.01	0.82
position	1.02	0.79	0.53	0.77	1.4	0.92	0.72	0.93
password	0.27	0.79	1.03	0.52	0.31	1.08	1.23	0.85
horoscope	1.02	0.54	0.78	0.52	1.5	1.16	1.24	1.02
motion_picture	0.27	0.79	1.03	0.52	0.84	1.32	1.36	1.13
photograph	0.27	0.79	1.03	0.52	0.65	1.11	1.16	0.89
baptism	0.78	1.04	0.29	0.52	1.22	1.41	0.75	1.02
automobile	0.78	1.04	0.78	1.02	1.1	1.45	1.18	1.42
quote	0.27	0.79	1.03	0.52	0.29	1.17	1.05	0.61
maid	1.02	0.79	0.53	0.77	1.21	0.86	0.53	0.92
remedy	1.02	0.79	0.53	0.77	1.14	0.87	0.74	1.01
vocal_music	0.02	0.04	0.03	0.02	0.54	0.41	0.36	0.37
yellow_pages	0.27	0.79	1.03	0.52	0.73	1.09	1.35	0.82
timeliness	0.02	0.04	0.03	0.02	0.16	0.39	0.28	0.55
buyer	1.02	0.76	0.51	0.77	1.47	1.16	0.73	1.1
caricature	0.27	0.79	1.03	0.52	0.79	1.09	1.27	1.31
marijuana	1.02	0.78	0.28	0.52	1.49	1.44	0.79	0.98
reproof	0.76	1.01	0.25	0.51	0.8	1.48	0.49	0.99
matter	1.02	0.54	0.78	0.52	1.23	0.54	0.8	0.85
population	0.77	1.04	0.78	1.02	1.3	1.32	1.2	1.54
Muses	0.27	0.79	1.03	0.52	0.48	0.79	1.13	0.72
voyage	0.77	1.04	0.78	1.02	1.12	1.45	1.13	1.35
marketplace	1.02	0.76	0.51	0.77	1.02	0.76	0.53	0.77
task	1.02	0.79	0.53	0.77	1.04	0.79	0.53	0.77
audience	0.27	0.79	1.03	0.52	0.43	0.95	1.22	0.52
paper	1.16	0.59	0.88	0.56	1.27	0.74	1.06	0.68
search	0.27	0.79	1.03	0.52	0.34	1.2	1.2	0.59
parts_of_flower	1.02	0.54	0.78	0.52	1.35	0.9	0.85	0.78

airline	0.77	1.04	0.78	1.02	1.21	1.31	1.08	1.46
preteen	0.02	0.04	0.03	0.02	0.12	0.04	0.19	0.31
plundering	1.02	0.76	0.51	0.77	1.07	1.06	0.76	1.03
wager	0.27	0.79	1.03	0.52	0.36	0.27	1.7	0.87
infant	0.02	0.04	0.03	0.02	0.51	0.34	0.28	0.41
ballot	1.02	0.79	0.53	0.77	1.4	1.07	0.79	0.78

### Appendix Three: Baseline Rankings

Query	Rankings			
	Altavista	Excite	Infoseek	Yahoo
canine	2	1	4	3
chatter	2.5	2.5	2.5	2.5
jest	3	2	4	1
frog	4	1	2	3
united_states_of_america	4	3	1	2
laughingstock	1	3	4	2
map	4	1	2	3
lyric	3	1	2	4
dictionary	2	1	4	3
weather	3	1.5	4	1.5
harmonics	4	1	2	3
wallpaper	2	3.5	1	3.5
lodgings	3	1.5	4	1.5
salutation	2	4	3	1
love	3	1	4	2
interval	3	2	4	1
friendship	2	3	4	1
position	1	4	2	3
password	4	1	2	3
horoscope	4	1	2	3
motion_picture	1	2	4	3
photograph	3	1	4	2
baptism	4	3	1	2
automobile	4	1	2	3
quote	3	1	4	2
maid	1	3	4	2
remedy	4	3	1	2
vocal_music	1	4	2	3
yellow_pages	2	4	3	1
timeliness	4	3	2	1
buyer	1	2	4	3
caricature	2	3	4	1
marijuana	3	4	1	2
reproof	4	2	3	1
matter	2	4	3	1
population	2	1	4	3
Muses	1	4	3	2
voyage	3	1	4	2
marketplace	3	3	1	3
task	1	3	3	3
audience	3	2	1	4
paper	4	1.5	1.5	3
search	4	1	2	3
parts_of_flower	3	1	4	2

airline	2	3	4	1
preteen	4	2	3	1
plundering	4	2	3	1
wager	4	1	3	2
infant	2	1	4	3
ballot	1.5	1.5	3	4

## Appendix Four: Initial Run Rankings

Query	Rankings				Error					
	Altavista	Excite	Infoseek	Yahoo	Altavista	Excite	Infoseek	Yahoo	sum	average
canine	1	3	2	4	-1	2	-2	1	6	1.5
chatter	4	2	1	3	1.5	-0.5	-1.5	0.5	4	1
jest	1	3	2	4	-2	1	-2	3	8	2
frog	1	3	2	4	-3	2	0	1	6	1.5
united_states_of_am	4	1	3	2	0	-2	2	0	4	1
erica										
laughingstock	2	1	4	3	1	-2	0	1	4	1
map	4	2	1	3	0	1	-1	0	2	0.5
lyric	4	2	1	3	1	1	-1	-1	4	1
dictionary	4	2	1	3	2	1	-3	0	6	1.5
weather	1	3	2	4	-2	1.5	-2	2.5	8	2
harmonics	3	1	2	3	-1	0	0	0	1	0.25
wallpaper	4	1	3	2	2	-2.5	2	-1.5	8	2
lodgings	4	1	3	2	1	-0.5	-1	0.5	3	0.75
salutation	1	3	2	4	-1	-1	-1	3	6	1.5
love	1	3	2	4	-2	2	-2	2	8	2
interval	4	1	3	2	1	-1	-1	1	4	1
friendship	1	3	2	4	-1	0	-2	3	6	1.5
position	1	2	4	3	0	-2	2	0	4	1
password	4	2	1	3	0	1	-1	0	2	0.5
horoscope	1	3	2	4	-3	2	0	1	6	1.5
motion_picture	4	2	1	3	3	0	-3	0	6	1.5
photograph	4	2	1	3	1	1	-3	1	6	1.5
baptism	2	1	4	3	-2	-2	3	1	8	2
automobile	3	1	3	2	-1	0	1	-1	3	0.75
quote	4	2	1	3	1	1	-3	1	6	1.5
maid	1	2	4	3	0	-1	0	1	2	0.5
remedy	1	2	4	3	-3	-1	3	1	8	2
vocal music	3	1	2	3	2	-3	0	0	5	1.25

yellow_pages	4	2	1	3	2	-2	-2	2	8	2
timeliness	3	1	2	3	-1	-2	0	2	5	1.25
buyer	1	3	4	2	0	1	0	-1	2	0.5
caricature	4	2	1	3	2	-1	-3	2	8	2
marijuana	1	2	4	3	-2	-2	3	1	8	2
reproof	2	1	4	3	-2	-1	1	2	6	1.5
matter	1	3	2	4	-1	-1	-1	3	6	1.5
population	4	1	3	2	2	0	-1	-1	4	1
Muses	4	2	1	3	3	-2	-2	1	8	2
voyage	4	1	3	2	1	0	-1	0	2	0.5
marketplace	1	3	4	2	-2	0	3	-1	6	1.5
task	1	2	4	3	0	-1	1	0	2	0.5
audience	4	2	1	3	1	0	0	-1	2	0.5
paper	1	3	2	4	-3	1.5	0.5	1	6	1.5
search	4	2	1	3	0	1	-1	0	2	0.5
parts_of_flower	1	3	2	4	-2	2	-2	2	8	2
airline	4	1	3	2	2	-2	-1	1	6	1.5
preteen	3	1	2	3	-1	-1	-1	2	5	1.25
plundering	1	3	4	2	-3	1	1	1	6	1.5
wager	4	2	1	3	0	1	-2	1	4	1
infant	3	1	2	3	1	0	-2	0	3	0.75
ballot	1	2	4	3	-0.5	0.5	1	-1	3	0.75
AVERAGE					-0.18	-0.18	-0.5	0.74		1.27

## Appendix Five: Post-User Feedback Run Rankings

Query	Rankings				Error					
	Altavista	Excite	Infoseek	Yahoo	Altavista	Excite	Infoseek	Yahoo	sum	average
canine	1	3	2	4	-1	2	-2	1	6	1.5
chatter	4	2	1	3	1.5	-0.5	-1.5	0.5	4	1
jest	1	4	2	3	-2	2	-2	2	8	2
frog	1	3	2	4	-3	2	0	1	6	1.5
united_states_of_america	3	2	4	1	-1	-1	3	-1	6	1.5
laughingstock	2	1	4	3	1	-2	0	1	4	1
map	4	1	2	3	0	0	0	0	0	0
lyric	4	2	1	3	1	1	-1	-1	4	1
dictionary	4	2	1	3	2	1	-3	0	6	1.5
weather	1	3	2	4	-2	1.5	-2	2.5	8	2
harmonics	4	1	2	3	0	0	0	0	0	0
wallpaper	4	1	3	2	2	-2.5	2	-1.5	8	2
lodgings	3	2	4	1	0	0.5	0	-0.5	1	0.25
salutation	1	4	2	3	-1	0	-1	2	4	1
love	1	1	3	4	-2	0	-1	2	5	1.25
interval	3	2	4	1	0	0	0	0	0	0
friendship	1	4	2	3	-1	1	-2	2	6	1.5
position	1	3	4	2	0	-1	2	-1	4	1
password	4	2	1	3	0	1	-1	0	2	0.5
horoscope	1	3	2	4	-3	2	0	1	6	1.5
motion_picture	4	2	1	3	3	0	-3	0	6	1.5
photograph	4	2	1	3	1	1	-3	1	6	1.5
baptism	2	1	4	3	-2	-2	3	1	8	2
automobile	4	1	3	2	0	0	1	-1	2	0.5
quote	4	1	2	3	1	0	-2	1	4	1
maid	1	3	4	2	0	0	0	0	0	0
remedy	1	3	4	2	-3	0	3	0	6	1.5
vocal_music	1	2	4	3	0	-2	2	0	4	1
yellow_pages	4	2	1	3	2	-2	-2	2	8	2

timeliness	4	2	3	1	0	-1	1	0	2	0.5
buyer	1	2	4	3	0	0	0	0	0	0
caricature	4	3	2	1	2	0	-2	0	4	1
marijuana	1	2	4	3	-2	-2	3	1	8	2
reproof	3	1	4	2	-1	-1	1	1	4	1
matter	1	4	3	2	-1	0	0	1	2	0.5
population	3	2	4	1	1	1	0	-2	4	1
Muses	4	2	1	3	3	-2	-2	1	8	2
voyage	4	1	3	2	1	0	-1	0	2	0.5
marketplace	1	3	4	2	-2	0	3	-1	6	1.5
task	1	2	4	3	0	-1	1	0	2	0.5
audience	4	2	1	3	1	0	0	-1	2	0.5
paper	1	3	2	4	-3	1.5	0.5	1	6	1.5
search	4	1	1	3	0	0	-1	0	1	0.25
parts_of_flower	1	2	3	4	-2	1	-1	2	6	1.5
airline	3	2	4	1	1	-1	0	0	2	0.5
preteen	3	4	2	1	-1	2	-1	0	4	1
plundering	1	2	4	3	-3	0	1	2	6	1.5
wager	3	4	1	2	-1	3	-2	0	6	1.5
infant	1	3	4	2	-1	2	0	-1	4	1
ballot	1	2	3	4	-0.5	0.5	0	0	1	0.25
AVERAGE					-0.3	0.1	-0.2	0.36		1.06

## Appendix Six: Improvement from Initial Run to Post-User Feedback Run

Query	Average Ranking Error			
	Initial Run	Post-User Feedback Run	Amount of Decrease	Amount of Change
canine	1.5	1.5	0	0
chatter	1	1	0	0
jest	2	2	0	0
frog	1.5	1.5	0	0
united_states_of_america	1	1.5	-0.5	0.5
laughingstock	1	1	0	0
map	0.5	0	0.5	0.5
lyric	1	1	0	0
dictionary	1.5	1.5	0	0
weather	2	2	0	0
harmonics	0.25	0	0.25	0.25
wallpaper	2	2	0	0
lodgings	0.75	0.25	0.5	0.5
salutation	1.5	1	0.5	0.5
love	2	1.25	0.75	0.75
interval	1	0	1	1
friendship	1.5	1.5	0	0
position	1	1	0	0
password	0.5	0.5	0	0
horoscope	1.5	1.5	0	0
motion_picture	1.5	1.5	0	0
photograph	1.5	1.5	0	0
baptism	2	2	0	0
automobile	0.75	0.5	0.25	0.25
quote	1.5	1	0.5	0.5
maid	0.5	0	0.5	0.5
remedy	2	1.5	0.5	0.5
vocal_music	1.25	1	0.25	0.25
yellow_pages	2	2	0	0
timeliness	1.25	0.5	0.75	0.75
buyer	0.5	0	0.5	0.5
caricature	2	1	1	1
marijuana	2	2	0	0
reproof	1.5	1	0.5	0.5
matter	1.5	0.5	1	1
population	1	1	0	0
Muses	2	2	0	0
voyage	0.5	0.5	0	0
marketplace	1.5	1.5	0	0
task	0.5	0.5	0	0
audience	0.5	0.5	0	0
paper	1.5	1.5	0	0
search	0.5	0.25	0.25	0.25
parts_of_flower	2	1.5	0.5	0.5

airline	1.5	0.5	1	1
preteen	1.25	1	0.25	0.25
plundering	1.5	1.5	0	0
wager	1	1.5	-0.5	0.5
infant	0.75	1	-0.25	0.25
ballot	0.75	0.25	0.5	0.5
AVERAGE	1.27	1.06	0.21	0.26

Kathryn Elizabeth Bechtold  
1341 Paulton Street  
Johnstown, PA 15905  
814.255.2014

## Professional Goal

To develop software for space exploration projects.

## Education

B.S. in Computer Science, with honors in Computer Science and Engineering  
The Pennsylvania State University  
16 December 2000  
3.96 GPA

**Honors Thesis:** An Analysis of the Accuracy of a Search Engine Ranking Algorithm for Meta-Search Engines Using the Summary Schemas Model

Thesis Supervisor: Dr. Ali Hurson

Honors Advisor: Dr. John Hannan

**Relevant Coursework:** Database Systems, Space Colonization, Artificial Intelligence, Software Design Methods, Programming Language Concepts, Computer Graphics, Data Structures and Algorithms, Numerical Computations, Applied Statistics in Science, Discrete Mathematics, Computer Organization and Design, Systems Administration, Business Data Communication, Technical Writing

## Honors and Awards

Schreyer/University Scholar	1996-2000
Kappa Theta Epsilon	1999-2000
Evan Pugh Scholar Award	1999 and 2000
Lockheed Martin Scholarship	1998
President Sparks Award	1998
President's Freshman Award	1997

## Research Experience

The Pennsylvania State University  
Dr. Ali Hurson  
Multidatabase Research Group  
August 1999 to December 2000

## Professional Organizations

Association for Computing Machinery  
Society of Women Engineers

## Professional Experience

5/00 - 7/00, **Software Engineer** at **NASA Glenn Research Center**

Optimized program data formats; developed virtual reality modeling tool; expanded web-based user interfaces.

1/99 - 8/99, **Software Engineer** (co-op) at **Lockheed Martin Mission Systems**

Corrected software defects; performed formal software tests; wrote system documentation.

8/96 - 5/99, **Computer Lab Assistant** at **Atherton Computer Committee** (Vice President, 1998)

Solved user problems; oriented new users; represented lab's interests at advisory councils.

1/98 - 4/98, **Project Consultant** at **Lucent Technologies**

Analyzed laser trench metrology process; formulated specific process improvements; worked with diverse team of student engineers.

## Technical Skills

**Programming Languages:** C, C++ (with OpenGL), HTML, VRML, SPARC assembly language, ML, LISP, Fortran 77

**Operating Environments:** UNIX, Linux, MS-DOS, MS Windows (all versions)

## Extracurricular Activities

Society of Women Engineers	1996-2000
Treasurer	1999-2000
Webmaster	1997-2000
Linux Users Group	1999-2000
Vice President	2000
Engineering Mentorship Program	2000
Technology Resource Group (Women in Engineering Program)	1999-2000
Web Developer and Writer	1999-2000
Atherton Executive Council	1998-2000
Public Relations Chair	1999-2000
Association for Computing Machinery	1999-2000
Philharmonic Orchestra	1996-1998
Schreyer Honors College Student-to-Student Mentor Program	1998-1999

