

Mining World Knowledge for Analysis of Search Engine Content

John D. King, Yuefeng Li, Xiaohui Tao, Richi Nayak
 {j5.king,y2.li,x.tao,r.nayak}@qut.edu.au
 School of Software Engineering and Data Communications,
 Queensland University of Technology, QLD 4001, Australia

Abstract—Little is known about the content of the major search engines. We present an automatic learning method which trains an ontology with world knowledge of hundreds of different subjects in a three-level taxonomy covering all the documents offered in our university library. We then mine this ontology to find important classification rules, and then use these rules to perform an extensive analysis of the content of the largest general purpose internet search engines in use today. Instead of representing documents and collections as a set of terms, we represent them as a set of subjects, which is a highly efficient representation, leading to a more robust representation of information and a decrease of synonymy.

Keywords: Ontology, hierarchal classification, taxonomy, collection selection, search engines, data mining

I. INTRODUCTION

Search engines have forever changed the way people access and discover information, allowing information about almost any subject to be quickly and easily retrieved. As increasingly more material becomes available electronically the influence of search engines on our lives will continue to grow.

The contents of the major search engines remain largely unknown to date. This research aims to change this. We introduce a new method which we use for the classification of large search engines, including those containing many billions of documents. Table I shows the search engines utilised in the present project. We compared the search engines across hundreds of subjects, and the similarities and differences between the engines were analysed. As far as the authors are aware this is the first time a study of this size and scope has been carried out.

Currently human experts are better at identifying relevant documents than the state of the art information retrieval methods. Human experts are also currently better at classifying

documents than the state of the art automatic classification methods. One factor that makes human experts superior from computer programs is ‘world knowledge’. World knowledge encompasses sophisticated contextual information on topics such as philosophy, psychology, religion, social sciences, language, natural sciences, mathematics, technology, the arts, literature, geography, and history. In this study we make use of world knowledge stored in an ontology. Ontologies have been used historically in Artificial Intelligence for a variety of applications. However, a major problem associated with building an ontology which covers a large number of domains is the human-hours that would be required to construct it. This problem is called the *knowledge acquisition bottleneck*. The aim of this research was to quickly, cheaply and simply build an ontology which has both a wide range of knowledge and capabilities across many different domains.

Representing collection descriptions as subjects, rather than terms has considerable advantages. For example, consider a computing dictionary from the 1970’s and a computing dictionary from today. They both cover the same subject, yet there are wildly different sets of terms within each dictionary. By representing collections by their subjects instead of their terms, a more robust system results, that is more adaptive in the face of technological and social change.

When we analysed the search engines, we specifically considered the subject distributions of each search engine. We found that some search engines had a bias towards the sciences and others toward the arts. The analysis also showed that Teoma and ASK use the same index for their results. Each search engine was also compared to Google revealing that AOL was most similar to Google (AOL uses a slightly different version of Google’s index) while WiseNut was the most different. Only results from the ten highest level subjects are presented due to space reasons, but our study covers hundreds of lower level subjects.

There are several motivations for this work. The first motivation is that little is known about the contents of the major search engines. The second is that the current methods for finding information about search engines do not cover a broad enough set of subjects to be useful when working with the major search engines. The third motivation is the difficulty of manually creating a large ontology for representation of world knowledge which is broad and deep enough to cover the subjects encountered in the major search engines. The fourth is that current methods for comparing search engines are not

Title	Abbreviation	URL
Altavista	AV	http://www.altavista.com/
America Online Search	AOL	http://search.aol.com/
Ask Jeeves	ASK	http://webk.ask.com/
Google	Google	http://www.google.com/
MSN Search	MSN	http://search.msn.com/
Teoma	Teoma	http://www.teoma.com/
WiseNut	Wisenut	http://www.wisenut.com/
Yahoo Search	Yahoo	http://www.yahoo.com/

TABLE I
THE SEARCH ENGINES USED IN THIS PAPER

able to find latent patterns of similarity between the search engines.

This multi-disciplinary paper therefore draws from the fields of collection selection, taxonomies, ontologies, ontology learning, data mining, and singular value decomposition. Each of these areas will be briefly outlined throughout this paper when relevant to the topic at hand. Three contributions are presented to the fields of ontologies, information retrieval (IR) and search engine analysis. The creation of a large ontology for representation of world knowledge for Web Intelligence is outlined. The second is the evaluation of popular search engines using both world knowledge and singular value decomposition. The third contribution is the method of selecting query probe terms from the ontology.

The rest of this paper is organised as follows. Section II introduces our automatic ontology learning method, Section III shows our analysis of the search engines, Section IV presents a formalisation of our methods, Section V gives our search engine analysis results, and Section VI concludes the paper.

II. AUTOMATIC ONTOLOGY LEARNING

This section will describe our selection and application of a method which allows us to automatically build a large ontology from a human classified training set. The problem with many existing ontologies is that they only cover a small number of domains, and each domain has to be manually added by a domain expert. The method presented in this section automatically creates an ontology covering hundreds of different domains. Automatic ontology learning will be a great improvement, enabling technologies to facilitate the creation of the semantic web.

A. Automatic Ontology Learning Introduction

There are three methods of ontology learning, each offering a trade-off between speed and accuracy. The three methods are:

- 1) to generate rules from free text (fast but inaccurate)
- 2) to generate rules from expert created and/or classified materials such as dictionaries and encyclopedia texts
- 3) ask domain experts to populate the ontology by manually entering rules (slow but accurate)

The second method is adopted in this research as it provides a balanced approach.

In this paper we present our automated ontology learning method called IntelliOnto for “Intelligent Ontology”. It differs from previous methods in that an taxonomic ontology based approach is used, and it also covers a large range of domains.

We generated the IntelliOnto ontology from a training set of over 80,000 documents. The training set is a large set of human expert classified documents covering many different subjects. This solved the problem of the time complexity it takes to get an expert to add rules to the ontology. We used statistical analysis such as support and confidence measures to extract classification terms from this training set. The IntelliOnto ontology generated is based on documents, so it can be partially classified as a *linguistic ontology*. However the documents are also classified in a taxonomy (or hierarchy) so

the IntelliOnto ontology can also be partially classified as a *tree based ontology*.

In its simplest form the IntelliOnto ontology consists of terms and subjects in the form of a *backbone taxonomy*. Each node in the taxonomy represents a subject. Each subject node can be represented by a description of the subject, a set of terms and term weights which are associated with the subject, and the inter-relations between it and other subject nodes. One term can be assigned to many different subject nodes; however if a term belongs to only a few nodes it is better for classification than if a term that belongs to many nodes.

B. Introduction to Subjects

While many information retrieval systems use terms¹ to describe documents and search engines, the IntelliOnto method uses subjects to describe documents and search engines. The power of a subject based approach is better understood through the following example. If a user types “matrix factorisation methods” into a search engine, they would expect “singular value decomposition” to be returned as a result. Both phrases belong to the same subject, yet there is no overlap of terms. By identifying the subject matter instead of using terms it is possible to return items where the terms do not overlap yet they are still highly relevant.

A subject can be difficult to define and abstract to some degree, while still containing a strong taxonomic structure. A subject may have sub and super subjects, with super-subjects being a higher abstraction of the subject. Many subjects can be defined by a domain vocabulary as can be easily observed if one compared the terms used in a movie review to the terms used in a computer science paper.

C. Ontology Construction Method Overview

This section will give a general overview of how we select and construct our IntelliOnto ontology. The stages involved in the IntelliOnto ontology construction process are:

- 1) Selecting a classification taxonomy
- 2) Identifying a training set
- 3) Downloading a training set and populating the IntelliOnto ontology
- 4) Cleaning up the ontology

Figure 1 shows the ontology building process. Included with each term are features such as the source document ID and the term frequency.

1) *Selecting a Classification Taxonomy*: Ideally there are several desirable properties in a good expert classified taxonomy. The taxonomy should cover a wide number of subjects, be carefully constructed, be standard across the world, and be available in different languages. It was decided to use the Dewey Decimal System, a widely used library classification system². The system has been used for the classification of a large number and wide range of materials. The Dewey

¹This paper follows standard Information Retrieval practise and refers to words or keywords as “terms”.

²For a full listing of the classifications see <http://www.tnrldlib.bc.ca/dewey.html>. For an example of the classification system in use see our university’s library web site. <http://libcat.qut.edu.au/>

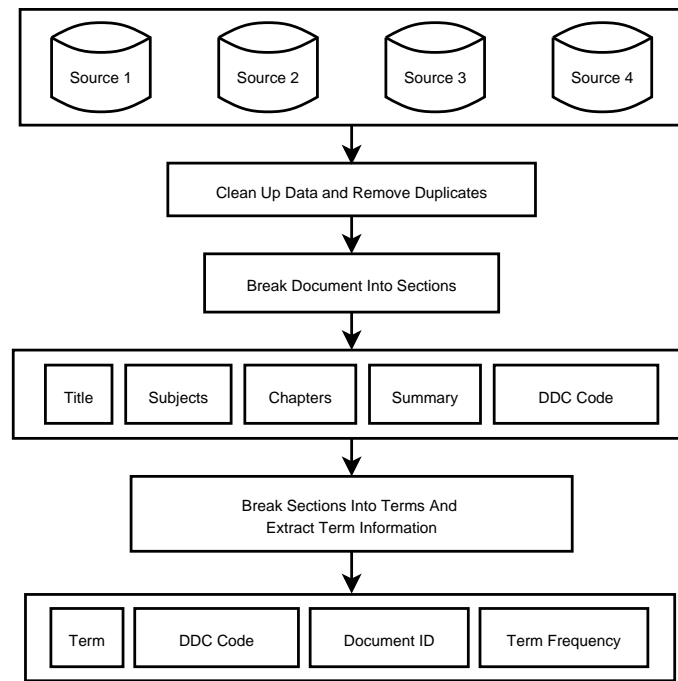


Fig. 1. Building The IntelliOnto Ontology Base

DDC	Subject Description
000	Generalities
100	Philosophy and Psychology
200	Religion
300	Social Sciences
400	Language
500	Natural Sciences and Mathematics
600	Technology (Applied Sciences)
700	The Arts
800	Literature and Rhetoric
900	Geography and History

TABLE II
THE TOP LEVEL OF THE TAXONOMY

taxonomy is based on a pattern of ten root nodes, with each root node having ten child nodes. Each child node has another ten child nodes with this pattern continuing downwards. There can be many different levels of the taxonomy, depending on how precise the subject match is. There are 1,110 classification nodes at the top three levels of the taxonomy, with many more nodes in the lower levels of the taxonomy. There are some low-level subject nodes that are unused because of depreciation or limited coverage. In this paper only the top three levels of the taxonomy are used.

Figure 2 shows part of the Dewey taxonomy, and Figure 3 shows a more detailed portion of the taxonomy. Each Dewey Decimal Code (DDC) provides the best possible classification for each item.

The top level contains general subject classifications, while the bottom level contains specific subject classifications. Each subject node covers the entire set of subject nodes below it. Moving down in the taxonomy focuses on a specific subject. Moving up in the taxonomy gives more general coverage. Table II show the subject descriptions of the ten root nodes of

the taxonomy.

Compared to many other web taxonomies such as the Open Directory Project (ODP)³, the Looksmart directory⁴, and the Yahoo directory⁵, it may be argued that the Dewey taxonomy is substantially better planned. Web directories are notorious for misclassified documents arising from sub-standard data handling that reflects a poor understanding on taxonomy. Another problematic issue with web directories is the presence of ad-hoc subject nodes. An advantage of the Dewey Decimal system is that the taxonomy is designed by trained classification experts, and almost all the additions are made by classification experts. The Dewey system is also multilingual, and it is possible to train the IntelliOnto ontology with other languages.

It would be possible to use other classification schemes (such as the Library of Congress Classification scheme⁶ or the Universal Decimal Classification Scheme) however the Dewey System was chosen because it is one of the more universally used schemes, it is better planned from a hierarchal perspective, and has more multilingual training data available. The authors also researched using WordNet as a training set, however there was too little training data available in the examples for it to be of use.

2) *Identifying a Training Set*: A human classified training set was chosen because automatic classification algorithms such as *k-means* clustering [38] are inaccurate and error prone. Our method used human classified materials and is thus more accurate.

The desirable properties of a training set are that it is large,

³<http://dmoz.org/>

⁴<http://www.looksmart.com/>

⁵<http://dir.yahoo.com/>

⁶<http://www.loc.gov/catdir/cpsolcco/lcco.html>

of high quality, and covers a wide range of subjects. A data set reflecting these requirements is the Queensland University of Technology Library Catalogue⁷, which contains over 80,000 usable items. This data set was used to populate the IntelliOnto ontology with world knowledge. Figure 4 shows an example item from the training set. Each document in our training set is assigned a Call Number. These documents have been carefully classified by experts in the field, and the information is of superior quality to other web based directories. However if anyone wishes to replicate this research, any university library with a wide and deep representation of knowledge would suffice as a training set.

3) *Downloading a Training Set and Populating the IntelliOnto Ontology:* For this research the entire Queensland University of Technology Library Catalogue was downloaded and parsed. Note that only items with a Dewey Decimal Code were used in this training set. The data extracted for each item includes the document title, chapter headings, description, notes, Dewey Decimal code⁸, subject groups, summary and description⁹. This metadata is treated as a compression of the document. Most of the metadata is rich in descriptive terms.

The raw training set data was then processed into a large set of low-level classification terms which were used as the base of the IntelliOnto ontology. Table III shows a sample of the base of the IntelliOnto ontology. This base is built by iterating through all the training set items from the library catalogue. Each item has a Dewey Decimal code associated with it. The metadata for each item such as title, chapter headings, summary and description is retrieved. Once the metadata for the item is retrieved from the library database it was parsed

⁷See <http://libcat.qut.edu.au/> This library web site is excellent for use as a training set because most of the entries have extra meta-information such as descriptions and summaries.

⁸Note that this system is not perfect, it is possible for an item to belong to more than one classification

⁹We do not actually index the document itself as only a few library documents are currently in full text format.

Term	DDC	Document ID	Frequency
approach	001.39	36	2
approach	101.1	46	1
interdisciplinary	001.39	36	1
psychoanalysis	001.40	37	1
domain	001.40	37	1
domain	001.70	39	1

TABLE III
A SAMPLE OF THE INTELLI ONTO ONTOLOGY BASE

Term	Part Of Speech	Count	ISF
history	NOUN	5910	-1.77664583141801
social	ADJ	4378	-1.47659199910067
congresses	NULL	4206	-1.43651207728051
book	NOUN	3166	-1.15246896145882
art	NOUN	3151	-1.14771986277514
study	NOUN	3041	-1.11218640869523
great	ADJ	2578	-0.947013904516996
language	NOUN	2547	-0.93491619599732

TABLE IV
TERMS WITH LOW INVERSE SUBJECT FREQUENCY. THESE ARE TERMS THAT OCCUR ACROSS SO MANY SUBJECTS AND HAVE SO MANY SENSES THAT THEY ARE OF NO USE FOR CLASSIFICATION.

into terms, and each term and it's Dewey Decimal code was saved in the IntelliOnto ontology.

With ontology learning, the more document metadata available for learning the better IntelliOnto performs. A feature of this IntelliOnto system is that if the user enters a query term that is not currently in the ontology, a *spider* can set out to retrieve relevant training data from other library catalogues in other libraries.

4) *Cleaning Up the IntelliOnto Ontology:* After the IntelliOnto ontology base was built, the data was cleaned up. Redundant data was removed to increase the efficiency of the ontology. This process involved removing stopwords, duplicate information, and unclassified information.

Stopwords are terms that are too common to be of any

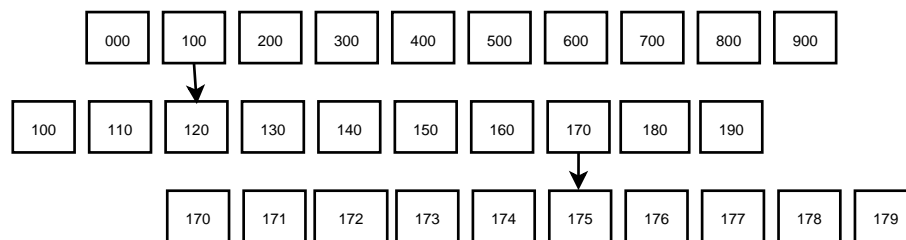


Fig. 2. The Dewey Decimal taxonomy

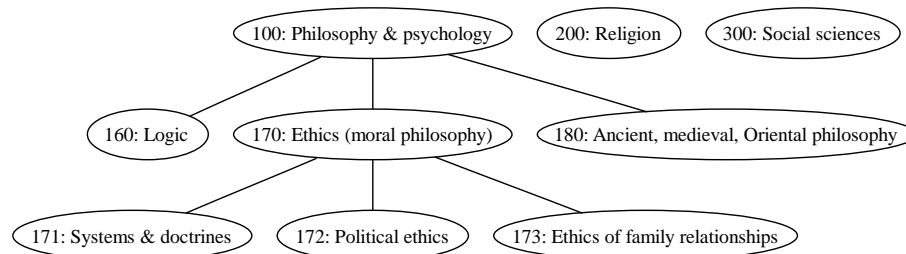



Fig. 3. A Portion of the taxonomy

Author	Neward, Ted.	
Title	Effective Enterprise Java / Ted Neward.	
Published	Boston : Addison-Wesley, c2005.	



ITEM LOCN	CALL NO	STATUS
Gardens Point	005.133 JAVA 228	IN LIBRARY

Ch. 1	Introduction	1
Ch. 2	Architecture	19
Ch. 3	Communication	101
Ch. 4	Processing	159
Ch. 5	State management	225
Ch. 6	Presentation	285
Ch. 7	Security	321
Ch. 8	System	379

Description xix, 470 p.: 23 cm.
 ISBN 0321130006 (pbk. : alk. paper)
 Summary "If you want to build better Java enterprise applications and work more efficiently, look no further. Inside, you will find an accessible guide to the nuances of Java 2 Platform, Enterprise Edition (J2EE) development. Learn how to: use in-process or local storage to avoid the network; set lower isolation levels for better transactional throughput; use Web services for open integration; consider your lookup carefully; pre-generate content to minimize processing; utilize role-based authorization; be robust in the face of failure; and employ independent JREs for side-by-side versioning."--BOOK JACKET.
 Subject Java (Computer program language)

Fig. 4. Example Training Set Page. Data extracted from this page includes the title, Call Number (Dewey Decimal Code), chapter titles and summary. Note that many of the terms are highly descriptive. Stopwords are discarded.

use in the classification process. Stopwords are removed from the IntelliOnto ontology to improve its performance. Most information retrieval systems use a static stopword list. However, our IntelliOnto system uses a dynamic stopword list which is created by identifying terms in the ontology which cover too many subject areas to be of any value. The *Inverse Subject Frequency* (ISF) metric (as defined in the Definitions section) was used for the novel purpose of identifying and removing these stopwords. Table IV shows the terms from our IntelliOnto ontology with the lowest Inverse Subject Frequency. Note that all of these terms will occur in a common English dictionary, yet they are of little or no use of identifying the subject matter of an item because they are so widely used. This leads to the notion that for a term to be included in a common English dictionary, it must be multi-purpose or reusable across different subjects.

D. Mining From the IntelliOnto Ontology

Once the IntelliOnto ontology base has been built from world knowledge, classification rules are mined from it. These rules are then used to classify collections such as search engines and databases.

There are many different classification rules that can be mined from the IntelliOnto ontology by using the terms, the subjects, and the taxonomy. By finding patterns between subject nodes and terms we are able to extract classification rules. These rules can then be made more useful by applying the taxonomic nature of the Dewey Decimal system.

When mining classification rules from the IntelliOnto ontology the correlation between the nodes of the taxonomy can be described in an association table. In its simplest form the association table shows the terms and the subjects that they belong to.

The classification terms need to be carefully selected. These terms should preferably be subject-specific (occurring within few or no other subjects) and should occur frequently within the subject and infrequently in other subjects. It is difficult to decide which terms to select as there are many possible terms to describe a subject. Many terms may not occur in common English dictionaries yet are still valuable for classification.

term	term count
software	281
programming	205
security	200
program	191
web	152
object	117
database	117
programs	105

TABLE V

TERMS THAT OCCUR MOST FREQUENTLY IN *005 Computer programming, programs, data*

These may include technical or subject specific terms such as conference names, acronyms and names of specialist technology. Some examples from computing are *RMI*¹⁰, *SMIL*¹¹, *XSLT*¹², and *servlet*¹³. Few standard English dictionaries include these terms, yet if any of these acronyms occur in a document it is likely the document covers a subject related to computing.

Our first term selection method, highest term frequency, involves selecting the most popular terms from each subject. Table V shows the most frequent terms for the subject *005 Computer programming, programs, data*.

Our second term selection method, highest support and confidence, involves finding the most distinguishing (or unique) terms from each subject based on confidence and support. Table VI shows the most distinguishing terms for the same subject. These terms cluster around the Dewey Decimal code "005". The nodes are grouped based on the third level of the taxonomy, any groupings below this level are not considered.

E. Results

Of the two ranking methods, the terms selected with high confidence and support thresholds were far better for search

¹⁰Remote Method Invocation.

¹¹Synchronized Multimedia Integration Language

¹²Extensible Stylesheet Language Transformation.

¹³"A Java application that, different from applets, runs on the server and generates HTML-pages that are sent to the client"
<http://www.softwareag.com/xml/about/glossary.htm>

Term	Count	Support	Confidence
c#	55	0.00003840	1
j2ee	48	0.00003351	1
javabeans	43	0.00003002	1
fedora	27	0.00001885	1
sax	27	0.00001885	1
awt	25	0.00001745	1
xsl	23	0.00001606	1
jdbc	23	0.00001606	1
oo	20	0.00001396	1
unicode	20	0.00001396	1

TABLE VI

TERMS FOR 005 *Computer programming, programs, data* WITH A CONFIDENCE SCORE OF ONE. NOTE THAT FEW OF THESE TERMS WOULD OCCUR IN A STANDARD ENGLISH DICTIONARY, YET THEY ARE EXCELLENT FOR CLASSIFICATION PURPOSES.

engine analysis than the terms selected by highest frequency. Some of the most frequent terms were so common across different subjects that they could virtually be considered stopwords. The results presented in this paper only use the highest confidence and support method.

F. Related Work

There is a growing body of work covering automatic and semi-automatic ontology learning. Automatic ontology learning has emerged as a separate entity from other ontology research, drawing from data mining, machine learning and psychology. However, automatic ontology learning is still very difficult to achieve other than in very specialised domains. We will briefly summarize some of the key research to date.

Maedche et. al. [40] presents methods for semi-automatically extracting ontologies from domain text. This includes methods for determining the measure of relationship between terms and phrases. Some ontology mining algorithms have been mentioned in [39], [41], which are the discoveries of the *backbone taxonomy* and the non-taxonomic relation.

Esposito et al. [11] provided semi-automatic ontology learning based methods for transforming raw text into a computer readable representation, enabling a computer to learn a language from training examples.

Faure et. al. [12] claims to have built a machine learning clustering system which learns subcategorization frames of verbs and ontologies from unstructured technical natural language texts. Unfortunately, in this example the methods were only tested within a single limited domain of cooking recipes which is itself highly structured (ie ingredients and cooking methods are fields common to all recipes).

Another prominent researcher in ontologies and library systems, Welty [54] uses description logics to allow reasoning and subject based classification within a library catalogue ontology. The user's search experience is improved by allowing search by description logic. In further work [53] he develops XML markup tagging for description logics with a library catalogue, an important development in the improvement of ontology reasoning. Weldt et. al. [55] also demonstrated how the use of an improved ontology can significantly improve information retrieval by 19%.

Buitelaar [2] selected 126 classification types and used WordNet as an ontology to assign almost forty thousand polysemic noun terms to one or more types in an automatically generated ontology. Each term could be disambiguated by what set of categories it belonged to or is excluded from. These groupings could then be used to tag corpora to aid automatic processing of data.

Suryanto et. al. [49] applied ontology learning to an existing well structured ontology allowing rapid extraction of rules. Kietz et. al. [28] applied semi-automatic ontology learning tools to a company intranet environment where natural language was mined for information.

Li et. al. [34], [35] presented a method of automatic ontology learning and refinement which can be used to model web user information needs. Stojanovic [48] used an ontology to refine search queries by removing term ambiguity. Queries were taken and mapped to their neighborhoods in a controlled vocabulary, then the neighborhoods were presented to the user for assessment. Gauch [50] uses hierarchal weighted ontologies to create a personalised user profile and to assist web browsing. The ontologies are used to classify web pages and user browsing habits into different categories, and the user profiles are matched to spidered web paged. Gandon [14] provided methods for managing distributed knowledge and assisting corporate activities by using ontologies.

The above references all contain examples of ontology generation and ontology learning. However many of the above examples use only a small, domain specific ontology with limited application. In this work we automatically create a large ontology covering hundreds of different domains.

III. SEARCH ENGINE ANALYSIS

This section will describe our development and application of a set of methods which allows us to analyse very large search engines across hundreds of different subjects by using the IntelliOnto ontology built in the previous section. Analysing an internet search engine without having direct access to it's index is difficult, particularly if the engine covers a broad range of subjects. In this research we analyse the content of eight of the largest and most popular search engines in use today.

A. Significance of Search Engine Analysis

There are many search engines distributed across the internet. *Search engine selection* is the selection of an optimal subset of search engines from a large set of engines for reducing search costs [3], [4], [9], [10], [13], [16], [17], [22], [37], [42]. A central aim of search engine selection is to accurately classify the content of each engine being evaluated. Once the content of each engine has been determined, the best subset of engines can be returned to serve an information need¹⁴.

For example, take two search engines, called *Search Engine A* and *Search Engine B*. Search Engine A contains information

¹⁴Many search engine selection methods require direct access to or communication with each engine, yet few internet search engines allow this. Thus other methods of evaluating search engine content must be developed.

on the *creative arts* and no information on *social science*. Search Engine B contains information on *social science* and less information on the *creative arts* than Search Engine A. Each engine is treated as a “black box” and no prior knowledge of the contents is assumed. A human expert is used to generate a set of significant classification terms for each subject. For the *creative arts* subject the set of classification terms may include *opera*, *ballet*, and *Mozart*. The set of terms which best classifies each subject is used to query each search engine and the number of times each term occurs in each search engine is recorded. Accordingly these results are then used to classify each search engine. It can be shown that Search Engine A is more suitable for finding information about the creative arts than Search Engine B, and any time a user requests information about the creative arts, Search Engine A can be returned as the best possible source for information.

As stated above, a search engine can be treated as a black box with no prior knowledge of its contents. All that is known is that when some information is sent, some information is returned from it. Based on what is returned some knowledge it gained about its contents. There are two main questions that need to be answered in order to find information about the contents of the black box.

- 1) Decide what to send to the black box?
- 2) Decide what to do with the information returned from the black box?

An ontology answers the first question. By using an ontology, we are able to refine information retrieval, by adding world knowledge to the system. By automatically generating an ontology, the problem of scalability is solved, allowing knowledge retrieval on hundreds of subjects. Thus the ontology helps select the best items to send to the black box by selecting the best classification terms from a wide range of subjects.

Taxonomy answers the second question. By transforming the probe term results into a taxonomy, a combined high-level and a detailed view of the information contained in the black box is achieved.

In this research this collection selection method is extended with a large-scale, taxonomy based, expert trained ontology called IntelliOnto which covers almost every domain of human endeavour. This ontology is mined to find significant classification terms for each subject. This eliminates the need to have domain experts assigning classification terms to each subject.

B. Methods of Search Engine Analysis

The following method is used for analysing the search engines using the IntelliOnto ontology:

- 1) Extract query probe terms from the IntelliOnto ontology
- 2) Query probe search engines with terms
- 3) Convert query probe results into taxonomy format
- 4) Perform singular value decomposition on query probe results

Figure 5 shows a representation of the search engine classification process.

1) *Extract Query Probe Terms from the IntelliOnto Ontology*: In collection selection, *query probing* [6], [8] is commonly used to discover the contents of uncooperative collections. Query probing involves sending a set of query terms to a collection and using the results to form conclusions about the collection’s content.

Subjects are used to classify search engines, as using a term histogram to describe each search engine is inefficient. Considering that the Oxford English Dictionary (Second Edition) contains 291,500 words [43], we estimated that a large general purpose search engine could have at least several times that number of terms in its index due to the many superfluous acronyms and technical terms that exist. This would necessitate sending a minimum of 291,500 queries to each search engine to assess its content distribution, and the same number of terms would be required to describe each engine. Therefore, because our search engine selection method used subjects to best describe each search engine, this means that large engines can be described with fewer than 1,000 subject classification codes. This is far more compact and robust than the standard search engine selection method of describing engines with a term histogram [4], [56].

Our method uses a *Lowest-Level-First* query probing approach. The query probe terms from each subject node of the lowest level of the taxonomy are extracted. While it was difficult to decide how many classification terms to extract for each subject node, the use of more terms allows better results for search engines which have a wider but more shallow coverage of a subject. However these engines may not have as high quality results as ones that provide deeper results for part of a subject. The use of fewer terms would result in better results for search engines which have a deeper coverage of some aspects of a subject but poor results for engines which have a wider coverage of a subject. In our experiments the top ten results from the highest confidence and support for each subject node are used.

2) *Query Probe Each Search Engine With Terms*: Once the query probe terms for each subject have been extracted from the IntelliOnto ontology they are sent to each search engine. The number of results for each term from each engine is extracted and saved¹⁵. The query probing algorithm is shown in Section IV.

In Table VII we show an example term-engine matrix. Each engine has a column (vector) to show the term frequencies. Terms can also be given weights of how important they are.

3) *Convert Query Probe Results Into Taxonomy Format*: Once the query probe terms have been sent to the search engine, and the results gathered, the terms need to be grouped into Dewey Decimal subject codes. To calculate the Dewey Decimal subject code results, the sum of the set of terms used to query probe the search engine for each Dewey Decimal subject is taken. For example, if ten terms from a subject are used to query probe a search engine, the results for each of the ten terms will be added together and this result recorded as the result for this subject code.

¹⁵We used an HTML parser for all the search engines except Google and Yahoo, where we used their respective APIs

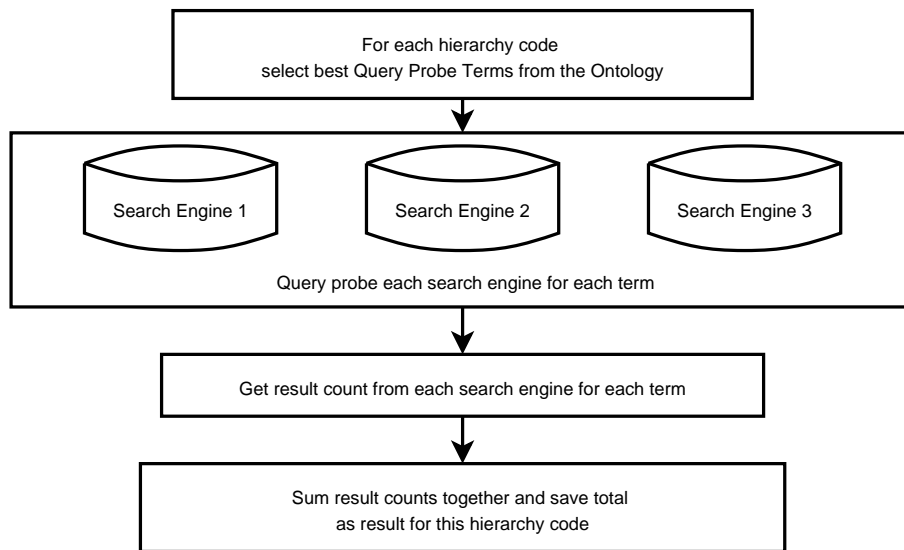


Fig. 5. Query Probing the Search Engines

term	Google	Teoma	AOL	MSN	ASK	AV	WiseNut	Yahoo
allusions	182000	558500	13867	530051	558400	3380000	52341	2710063
almanac	1770000	6287000	105334	2556384	6287000	34300000	527202	33109861
almanacs	413000	1540000	24934	451627	1540000	11400000	151520	7982030
almodovar	131000	362700	8867	225446	362600	2860000	8799	2599082
alofa	45800	45100	3314	255593	45000	461000	13071	326749
alphabet	2170000	6874000	152667	6856957	6874000	41300000	1492067	37867497
alphabetically	3000000	7072000	169334	6514059	7072000	34900000	1358432	30532537
alte	1080000	4366000	86001	54102142	4366000	89200000	276579	65999399
alternate	7730000	17810000	532667	12413491	17760000	103000000	1919870	81289554
alternately	620000	2288000	46534	1463828	2281000	9290000	24079	7901831
amateurism	15900	49200	1034	35769	49100	312000	790	236152
amateurs	885000	2703000	70667	71853644	2697000	63800000	271825	18094720

TABLE VII
THE TERM-FREQUENCY TABLE FOR EACH SEARCH ENGINE

The following formula is used for each subject in each search engine (where $k=10$):

$$\sum_{i=1}^k count(term_i) \quad (1)$$

The following algorithm shows the process for calculating the Dewey Decimal code results:

- 1: **for** Each search engine **do**
- 2: **for** Each top-level subject from DDC code 000 to 999 **do**
- 3: Calculate the result value for each subject node using equation 1;
- 4: Write the subject node value to the result table;
- 5: **end for**
- 6: **end for**

Table XI shows a sample of the results of probing each search engine with the set of terms with highest confidence and support for each Dewey Decimal code. There are many more results which we cannot show here because of space limitations. The Dewey Decimal codes in the table stand for:

- 003 Systems
- 004 Data processing Computer science
- 005 Computer programming, programs, data
- 006 Special computer methods
- ...
- 300 Social sciences
- 301 Sociology & anthropology
- 302 Social interaction
- 303 Social processes
- 304 Factors affecting social behavior

The results are then grouped together and the top ten levels of each search engine are calculated and the results presented in the Results section at the end of this paper.

4) *Perform Singular Value Decomposition on Query Probe Results:* We perform singular value decomposition (SVD) [29], [31] on the data from Table XI to analyse the results of the experiments. Singular value decomposition is a matrix factorisation and dimension reduction method that has many uses such as information retrieval; time series analysis; stock market analysis; and pattern matching. In this research, SVD is used for determining how related the subject matter of each search engine is to each of the other search engines. The advantage of SVD is that it can quickly show the latent

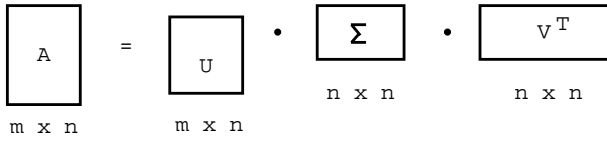


Fig. 6. Matrix Decomposition

relationship patterns between the search engines, and this method is able to identify patterns in the results which are difficult for a human to see. The result of the SVD calculation is a square “search engine-search engine” matrix with values between 1.0 and -1.0. In our case, a score of one means an exact match of content between the search engines and a score of zero means that there is no overlap of content between the search engines.

Properties of Singular Value Decomposition

Figure 6 shows the decomposition of the $m \times n$ matrix A , where $m \geq n$, and A is a matrix that represents search engines such that the rows are terms and columns are search engines (i.e. vectors). The *singular value decomposition* of A is said to be the factorisation:

$$A = U \Sigma V^T \quad (2)$$

where the diagonal of Σ is said to be the singular values of the original matrix, A :

$$\Sigma = \begin{bmatrix} w_0 & 0 & 0 & 0 \\ 0 & w_1 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & w_{n-1} & 0 \\ 0 & 0 & 0 & w_n \end{bmatrix}$$

and

$$U^T U = V^T V = I \quad (3)$$

$$w_1, w_2, \dots, w_{n-1}, w_n \geq 0 \quad (4)$$

In Equation 2, matrices U and V are orthogonal. The columns of U are called left singular values (terms) and the rows of V^T are called right singular values (search engine vectors). In this research the left singular values which are also commonly represented as U are ignored.

The orthogonal matrices V and U are formed by the eigenvectors of the matrices $A^T A$ and AA^T respectively. The dimensions of A are reduced to a smaller set of eigenvectors that are closest to the original matrix. This dimension reduction produces a clustering effect in the reduced matrix, removing noise from matrix and bringing together related concepts.

The Meaning Of Singular Value Decomposition

When singular value decomposition is applied to a matrix, the lesser patterns and background noise are removed, revealing the main patterns of the matrix. This is achieved by selecting the highest singular values, and reducing the lengths of the vectors in space. The subject column also becomes transformed as a vector in this reduced space, and lies close

to similar search engines. This can have the effect of bringing related eigenvectors closer to other eigenvectors which do not use the same terms. The closeness of the patterns of occurrence of subjects with similar values is what provides the latent pattern matching properties.

C. Related Work

People depend heavily on general purpose search engines to provide their information, yet there is no measure of what kind of information is provided by each engine, or what bias may be present. We will briefly summarize some of the key research to date in this section.

1) *Collection Selection*: The closest thing to search engine content analysis is the study of *collection selection*. *Collection selection* is the matching of a set of related collections with an information need.

The problems of collection selection have been addressed in previous work such as CORI [4] and GIOSS [17]. CORI assumes the best collections are the ones that contain the most documents related to the query. GIOSS uses a server which contains all the relevant information of other collections. Users query GIOSS which then returns an ordered list of the best servers to contact to send the query to. In a comparison of CORI and GIOSS [9] it was found that CORI was the best collection selection method, and that a selection of a small number of collections could outperform selecting all the servers and a central index.

Web based collection selection introduces its own set of problems, in that there is usually no direct access to a collections statistics, and that there is rarely cooperation between the collections and the collection broker. Our previous work [29], [31] in web based collection selection used query sampling methods that did not require communication with the broker or metadata about each collection. Singular value decomposition was then used on the results of the queries to select the best collection. These techniques were tested on the INEX collection with satisfactory results. In other work [51], a subject based approach was used to information fusion and was found to be promising and efficient. In [30] a short preview of the work presented in this paper was presented.

Si et. al. [45] present a web based modification of CORI called *ReDDE* which performs as well as or better than CORI by using an collection size estimate to supplement selection. They introduce an collection size estimation technique which is more efficient than in other estimation techniques such as the capture-recapture method [36].

Hawking et al [23] presented a method which used both centralised and distributed collection selection techniques. They also made use of anchor text to extract information on collections that have not been indexed.

Si et. al. [46] presented a method for minimalising the poor quality results returned by collections which have not implemented good information retrieval methods. By including the retrieval performance of each collection in the collection ranking, this problem can be reduced. A method for approximating the retrieval effectiveness of a collection, known as RUM, was presented. The RUM method was compared

to CORI and outperformed CORI in all the experiments conducted.

A common problem with traditional collection selection techniques are that they require communication between the search broker and collections, or that they need topical organisation. In this paper we presented a form of collection which does not need communication between the search broker and collections, and does not need topical organisation, and we applied this method to search engine analysis.

2) *Query Probing*: Web based collection selection commonly uses *query probing*, which involves sending query terms to a collection and analysing the results in order to find the best collection for an information need. While some collection selection techniques require the implementation of cooperative interfaces between the search broker and the server, query probes do not require special protocols between broker and server. The two main types of query probing are *Query Based Sampling* [5] and *Hierarchal Probing*¹⁶. Query Based Sampling involves sending random terms to a collection until it returns a document. The terms from the found documents are then used for further queries of the search engine until a threshold number of documents is reached. In Hierarchal Probing [24], a set of queries for the top level of the hierarchy are generated and sent to the collection. This continues down the hierarchy until a threshold limit number of returned documents is reached. A content summary for the engine and the place of the engine in the hierarchy is returned.

Callan [6] proposed that only three hundred query probes are needed to evaluate the contents of a collection, however we found that it takes many more query probes to accurately assess the content distribution of a large general purpose collection because of the broad range of subjects they cover. In the same work Callan presented a system that learnt a content summary through query probing a search engine. In other work Callan et. al. [5] postulated that “increases in document sample size do not tend to result in comparable improvements in content summary quality”.

Ipeirotis et. al. [26] trained a document classifier, generated rules from the classifier, then used these rules to perform query probing. The probes that returned the highest number of matches were then used to classify the collection. They [24] also used hierarchal “focused query probes” to adapt to the collection contents and try to find the depth of the collection, and estimated document frequencies for each term. Information was stored in a content summary for each collection as a result. However they also argued that because of Zipf’s law, query probing cannot find many low-frequency terms in an collection, which leads to incomplete content summaries. Further, that since collections that are topically similar often share the same lexicon, they share content summaries across topically similar collections. They hierarchically categorised and used smoothing for the content summaries to improve content summaries and collection selection.

Panagiotis et. al. [25] trained a system with a set of documents pre-classified into taxonomy topic areas. They then selected the terms which best defined each topic using a

selection algorithm and generated a set of topic rules. Finally, the topic rules were used for query probing, however only the number of items returned was used, the results themselves were discarded.

Gravano et. al. [18] used a document classifier for query probing collections. The authors used machine learning to generate document classifiers, followed by creating rules from the classifiers. The system only used the number of returned results, rather than the actual results. Additionally, they defined coverage and specificity and applied them when selecting which place in the taxonomy to assign a collection.

The previous references show the past work on query probing for the purpose of collection selection. Our work made use of and extended this research for the purpose of search engine analysis.

3) *Search Engine Analysis*: Most previous search engine analysis research involved evaluating search engines using metadata in such areas as size, change over time, overlap, and usage patterns.

In 1998 Lawrence et. al. [32] analysed the coverage of search engines in proportion to the total size of the web. They found that even the largest general purpose search engine covered fewer than one-third of the total web. Unfortunately the World Wide Web changes and grows so fast that surveys such as these become quickly outdated.

In the field of search engine performance evaluation [21], [33] Hawking et. al. [19] compared search engines using web search query logs and methods learned from TREC¹⁷. Hawking et. al. [20] further compared the retrieval performance of eleven search engines based on usefulness of search results for finding online services. Chowdhury et. al. [7] compared search engines using known search results, a set of web search query logs, and a corpus with relevance judgements. Beitzel et. al. [1] uses ODP and Looksmart along with a set of web search query logs to evaluate search engines. Gordon [15] and Chowdhury [7] show that some search engines perform better than others for some queries. However, overall the search engines returned statistically similar results.

Spink et. al. [47] analysed how users search a major search engine. They found that most searchers use the shortest path route, using few query terms, making few query changes, and viewing few query result pages. Jansen et. al. [27] confirms this study. Zwol [52] evaluated the usability and retrieval effectiveness of major search engines concentrating on user satisfaction.

The previous references shows the past work in the field of search engine analysis. However none of them perform a full analysis of the content of large, general purpose search engines. Our work extended this work by doing a large scale analysis of the largest search engines in common use today.

IV. FORMALISATION OF INTELLiONTO ONTOLOGY

In this section the formalisation of the proposed IntelliOnto Ontology is presented along with the related automatic ontology learning methods. Firstly the ontology structure is described, followed by the lexicon level of term-subject matrix

¹⁶Otherwise known as *Focused Probing*

¹⁷TREC stands for Text REtrieval Conference. Website <http://trec.nist.gov/>

of the proposed ontology. Next the first step of ontology learning is introduced for how the candidate terms representing a subject are selected. Finally the ontology is built based on the relationships existing in the subjects.

A. The Ontology Structure and Term-Subject Matrix

Definition 1: Let *OntoBASE* be the ontology base of the taxonomy, the ontology base is formally defined as a 2-tuple $OntoBASE := \langle S, R \rangle$, where

- S is a set whose element is called subject;
- R is a hierarchical set whose element is called relation.

Definition 2: A subject s in the subject set $S := \{s_1, s_2, \dots, s_n\}$ is a 3-tuple $s := \langle code, termset, abs \rangle$, where

- *code* is a unique identification code assigned by the Dewey Decimal Code system to the subject s ;
- *termset* is a set of terms representing the subject s ;
- *abs* is an abstract name of the subject s .

Regarding to each s in this paper we denote the taxonomy code of a subject s by $code(s)$, denote the set of terms representing s by $termset(s)$, and the name of s by $abs(s)$.

Definition 3: A relation r in the Relation set $R := \{r_1, r_2, \dots, r_m\}$ is a 3-tuple $r := \langle type, x, y \rangle$, where

- *type* is a set of relationship types, which has two elements $type := \{kindOf, partOf\}$;
- x and y are the subjects or terms that hold the relation r .

Usage: $\langle kindOf, s_1, s_2 \rangle$ means *subject s_1 is a kind of s_2* . And $\langle partOf, t_1, s_1 \rangle$ means *term t_1 is a part of subject s_1* . Regarding to r , in this paper we will denote the *type* of r by $type(r)$, and x by $x(r)$, and so as $y(r)$.

Definition 4: Let O be the proposed ontology. The ontology structure is formally defined as a 4-tuple $O := \langle S, H(S), R, \sigma \rangle$, where

- S is a set of subjects as defined in *OntoBASE*;
- $H(S)$ is a subject hierarchy. and $H(S) \subseteq S \times S$;
- R is a set of relations as defined in *OntoBASE*;
- σ is called signature mapping ($\sigma : S \rightarrow 2^S$) that defines the set of children of a given subject;

A lexical level for the ontology named *term-subject matrix* is defined as a quadruple and described as follows:

Definition 5: A term-subject matrix $M(O)$ in the ontology structure $O := \langle S, H(S), R, \sigma \rangle$ is a quadruple $M(O) := \langle T, S, TS, \eta \rangle$, where

- T is the set of terms assigned to all subjects S ;
- TS is a $m \times n$ zero-one matrix, where $n = |T|$ and $m = |S|$. For example, $TS(t_i, s_j) = 1$ means term $t_i \in termset(s_j)$, and $TS(t_i, s_j) = 0$ means $t_i \notin termset(s_j)$;
- η is called reference, a mapping ($\eta : T \rightarrow 2^S$), that defines the associated terms to a subject. For a term $t \in T$

$$\eta(t) = \{s \in S | TS(t, s) = 1\} \quad (5)$$

and its reverse is a set of terms, which satisfies

$$\eta^{-1}(s) = \{t \in T | TS(t, s) = 1\}. \quad (6)$$

Based on the term-subject matrix $M(O)$, one term may refer to multiple subjects, and one subject may be referred to by

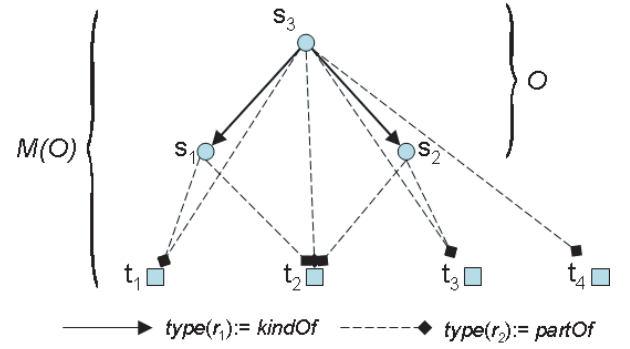


Fig. 7. A Simplified Sample of the Proposed Ontology, where s_i denotes a subject and t_j denotes a term.

	s_1	s_2	s_3
t_1	1	0	1
t_2	1	1	1
t_3	0	1	1

TABLE VIII

A SIMPLIFIED TERM-SUBJECT MATRIX

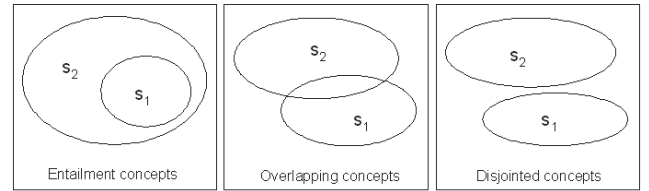


Fig. 8. Various Situations of Conceptual Areas Referred by the Different Subjects

multiple terms. Table VIII illustrates a simplified sample of term-subject matrix. Given a term t_1 , a set of relevant subjects $\{s_1, s_3\}$ that t_1 refers to can be identified. On the other side, given a subject s_2 , a set of relevant terms $\{t_2, t_3\}$ that s_2 is referred by can be identified as well. By using the term-subject matrix 4-tuple $M(O) := \langle T, S, TS, \eta \rangle$, the sample matrix can form a sample ontology which is illustrated in Figure 7. One may see in the figure that the subject s_3 is the parent of s_1 and s_2 , as all the terms including those referring to s_1 and s_2 refer to s_3 . s_3 covers broader conceptual area than s_1 and s_2 .

With the term-subject matrix, the proposed ontology can be further defined as a pair of $(O, M(O))$, where O is the ontology structure and $M(O)$ is the term-subject matrix. Given two subjects s_1 and s_2 , if $termset(s_1) = termset(s_2)$, we may say that $s_1 = s_2$. If $termset(s_1) \subset termset(s_2)$, we say that s_1 is a *kind-of* s_2 , since every term referring to s_1 also refers to s_2 , but s_2 has more term referred. The conceptual area referred by s_1 is entailed in s_2 in this case. If $termset(s_1) \cap termset(s_2) \neq \emptyset$ and $s_1 \neq s_2$, we may say that the conceptual areas referred by s_1 and s_2 are overlapping. However, if $termset(s_1) \cap termset(s_2) = \emptyset$, the conceptual areas referred by s_1 and s_2 are disjointed. The situations of containing/contained, overlapping, and disjointed concept areas are illustrated in Figure 8.

B. Assigning Candidate Terms to a Subject

Let $|D|$ denote the length of the training document set D . Each document $d \in D$ is represented by a set of terms $termset(d)$, $df(t)$ is the number of documents in D with $t \in termset(d)$, and $sf(t)$ is the number of subjects in S with $t \in termset(s)$. Instead of the traditional *inverse document frequency(idf)* [44] presented as Equation (7), we introduce *inverse subject frequency(isf)* presented as Equation (8):

$$idf(t) = \log\left(\frac{|D|}{df(t)}\right) \quad (7)$$

$$isf(t) = \log\left(\frac{m}{sf(t)}\right) \quad (8)$$

isf shows terms that occur across too many different subjects to be of no use. The inverse subject frequency method is used for term pruning. Terms with low isf value are considered “stopwords” and are subsequently removed from the ontology base.

A term-subject pair $p(t \rightarrow s)$ in $M(O)$ with their confidence and support values is referred to as a pattern $p(t \rightarrow s) := \langle t, s, conf(t \rightarrow s), sup(t \rightarrow s) \rangle$ in this paper, where $t \in T, s \in S, conf(t \rightarrow s) = [0, 1]$ and $sup(t \rightarrow s) = [0, 1]$. We use a modified support and confidence method for our system, in order to accommodate the taxonomy. The $conf(t \rightarrow s)$ and the $sup(t \rightarrow s)$ in the pattern describe the extent to which the pattern is discussed in the training set. The $conf(t \rightarrow s)$ and $sup(t \rightarrow s)$ are defined as follows:

$$conf(t \rightarrow s) = \frac{sf(t, s)}{sf(t)} \quad (9)$$

$$sup(t \rightarrow s) = \frac{sf(t)}{n} \quad (10)$$

where $sf(t, s)$ is the number of child subjects under s (including s) with t occurred in the $termset$. The greater $sup(t \rightarrow s)$ and $conf(t \rightarrow s)$ are, the more important the term t is to the subject s .

The following algorithm shows the process for selecting the candidate terms which represent the a subject and generating the rules which specifies the level of confidence of the candidate terms. Only the top 10 terms with the highest confidence and support values are selected as the candidates in the $termset(s)$ to represent the subject s . To avoid rare terms and spelling mistakes only terms that occur more than twice in the ontology are used for the generation.

- 1: **for** each $s \in S$ **do**
- 2: let $P = \{p | p = (t, s, conf(t \rightarrow s), sup(t \rightarrow s)), t \in T\} // P$ is the pattern set of s ;
- 3: sort P by $conf(t \rightarrow s)$ values in descending order;
- 4: **for** each group of the patterns with the same $conf(t \rightarrow s)$ value **do**
- 5: sort the patterns by $sup(t \rightarrow s)$ values in descending order;
- 6: **end for**
- 7: **for** ($n = 0, n < 9, n = n + 1$) **do**
- 8: let $p = (t, s, conf(t \rightarrow s), sup(t \rightarrow s)), t \in T$ be the top pattern of P ;
- 9: $TS(t, s) = 1$;

Subject	Term set	Terms
s_1	$termset(s_1)$	{computer, information, system}
s_2	$termset(s_2)$	{system, organisation}
s'_k	$termset(s'_k)$	{information, technology, system}
s_k	$termset(s_k)$	{computer, information, system, organisation, technology}

TABLE IX
SUBJECT EXAMPLES

- 10: $P = P - \{p\}$;
- 11: **end for**
- 12: **end for**

C. Build Taxonomy from Lowest Level

Based on Definition 5, for any child subject s_c on the lower level of taxonomy and its parent subject s_p on the upper level, we may have $termset(s_c) \subset termset(s_p)$. For any subjects on the lowest level which have no child, its candidate term set may be it’s final term set. However, for the subjects on the upper levels which have children, we need another method. The final terms assigned to $termset(s_p)$ may consist of its children subjects’ term sets and its own candidate term set, which may be formalized as:

$$termset(s_p) = \left(\bigcup_{\langle kindOf, s_c, s_p \rangle} termset(s_c) \right) \cup termset(s'_p). \quad (11)$$

where $termset(s'_p)$ is the candidate term set assigned to s_p by using the algorithm introduced in Section (IV-B). If a subject s is on the lowest level and has no child, the $termset(s) = termset(s')$. The $termset(s)$ that the subject s consists of would be only its own $termset(s')$. Table IX illustrates an example of that, where s_k is the parent of subjects s_1 and s_2 , $termset(s'_k)$ is the candidate term set assigned to subject s_k by the valuable patterns, and $termset(s_k)$ is the final term set representing the subject s_k .

V. RESULTS

This section presents the results of our search engine analysis. We mined the IntelliOnto ontology to extract terms which we used to analyse eight of the largest search engines in common use today. As exactly the same set of terms are sent to each search engine we can compare differences *within subjects*, ie comparing difference between Google and Yahoo for the Dewey Decimal code 100. However, note that differences in content *between subjects* cannot be accurately compared, ie between Dewey Decimal Code 100 and Dewey Decimal Code 200 for Google. This is because it is possible that highly subject specific terms exist in Google but do not exist in our classification set.

Our final results were:

- Terms with highest confidence and support produced better results than terms with highest frequency because highest frequency terms tend to be used across too many subject nodes to be of use for classification. However terms which occurred frequently within only one subject were excellent for classification.

	Google	Teoma	AOL	MSN	ASK	AV	Wisnut	Yahoo
Google	1.000							
Teoma	0.988	1.000						
AOL	0.990	0.989	1.000					
MSN	0.970	0.970	0.941	1.000				
ASK	0.988	1.000	0.989	0.970	1.000			
AV	0.962	0.981	0.973	0.936	0.981	1.000		
WiseNut	0.878	0.869	0.919	0.762	0.869	0.860	1.000	
Yahoo	0.974	0.971	0.943	0.991	0.971	0.948	0.769	1.000

TABLE X

SINGULAR VALUE DECOMPOSITION RESULTS FROM HIGHEST CONFIDENCE AND SUPPORT QUERY PROBES. A VALUE OF 1.0 INDICATES AN EXACT MATCH.

DDC	Google	Teoma	AOL	MSN	ASK	AV	Wisnut	Yahoo
003	5828103	22527390	394555	17579547	22526590	109051700	2196731	103908804
004	124621000	423407800	8258274	586610946	423571000	607970000	37710197	2021756676
005	307610000	1429860000	28731341	1164203725	1429600000	7802000000	141364738	4524730458
006	9966410	30818550	712167	52045800	30821950	232032600	3221223	210318846
...
300	495430	1138090	32095	1605376	1138090	12453836	97634	10454182
301	1058254	2413826	73255	3663311	2413757	19351710	282140	17553722
302	5575510	12082899	360010	11085727	12080899	71339032	1215382	65757962
303	3077000	11948500	208740	7378490	11925600	66210000	738114	59831238
304	1335546	5785801	325699	8569204	5786801	50394365	644650	46048934

TABLE XI

DETAILED RESULTS FROM HIGHEST CONFIDENCE AND SUPPORT QUERY PROBES.

- Altavista, Ask and AOL are censoring certain terms.

Table X shows the singular value decomposition of the results of the lowest level of the taxonomy for each search engine. An example of the lowest level data can be seen in Table XI. Each value in the table has a value between -1.0 and 1.0, with a 1.0 indicating an exact match of the data. SVD was used because it is able to find latent patterns between different sets of data. This table shows that:

- The content distribution of the different search engines was similar. We hypothesize that a good general purpose search engine will tend to reflect the content distribution of the surface World Wide Web.
- Teoma and ASK use the same index for their results. This can be seen in Table X because the number 1.0 shows an exact match of data in the intersection cell of ASK and Teoma.

Figure 9 shows a comparison of Google to the other search engines using singular value decomposition. This figure shows that:

- The search engine that was most similar to Google was AOL, which uses Google's results in its index. We suspect that the reason the SVD for Google and AOL was not the same is that Google is using a more recent index than AOL
- The search engine that was the most different to Google was WiseNut

Because of space limits only the total results of the top level of the taxonomy are shown. After calculating the lowest level of the taxonomy, the results are grouped together to view the top level of the taxonomy. For example, for the top level Dewey code 100 the sum of the results of Dewey codes 101 to 199 are taken. Table XII shows the raw set of results obtained

from each search engine for the upper level of the taxonomy. It can be seen that some search engines favour some subjects over others.

It can be seen that Table XIII shows the normalised results for highest confidence and support query probes, as well as the mean and the variance.

Figure 10 shows the content distributions of the search engines for each of the top level Dewey Decimal subject groupings. This figure shows that bias of each search engines.

It was also found that terms which occur in standard dictionaries tend to be multipurpose and were not much use for classification, while terms that are good for classification are single purpose and were generally too specialised for inclusion in standard English dictionaries.

VI. CONCLUSION

A novel form of ontology based search engine selection, IntelliOnto, which is scalable and easily distributed was introduced. This line of research was extended by performing search engine analysis of the largest general purpose search engines in use today. To do this we generated a set of classification queries from a large ontology populated with world knowledge, and each query was sent to each search engine. The results were then transformed into taxonomy groupings. The search engines' coverage in each of the top ten subject nodes of a large taxonomy was shown. An interesting observation was how similar the content distribution of the different search engines were to each other (and to Google). Our hypothesis is that a good general purpose search engine will tend to reflect the content distribution of the surface web.

This method is still experimental and with any work of this scope there is a margin for error, especially with the large number of heterogenous subjects contained in large general

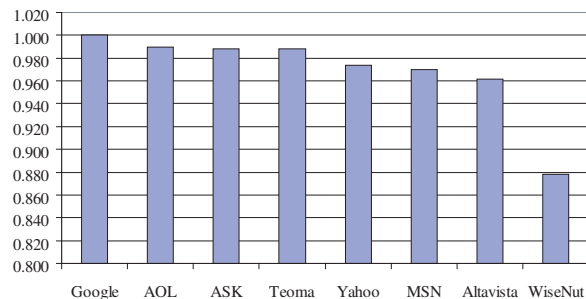


Fig. 9. Comparison of Google To Other Search Engines Using Singular Value Decomposition

DDC	Google	Teoma	AOL	MSN	ASK	AV	Wisnut	Yahoo
000	527475825	2099809634	43459545	2040592835	2099549734	9985621059	206583253	7994546979
100	52603205	160593582	3714695	306524615	159415702	1394522727	15026901	1242163522
200	14558441	53935067	997744	62265462	53891157	339809930	4569518	303515592
300	35786394	94711096	2574393	133150455	93570337	690771343	10647135	599482934
400	216564134	987020021	14895848	1264660410	986724552	5310352883	8867746	4905599616
500	129921685	431603076	8717298	466085409	431504526	2375110932	36846935	2182558297
600	123920874	340515770	8583559	447882212	340377470	2334727600	40272167	2121065851
700	176562186	730736766	12101745	1051602209	730613987	5053329192	69261591	4499093548
800	109177265	371954729	7190826	447786710	371833467	2492032242	29726841	2217022126
900	118071282	398285674	8102017	457928597	398053309	2730368118	38607531	2489312595

TABLE XII

RAW RESULTS FROM HIGHEST CONFIDENCE AND SUPPORT QUERY PROBES. EACH DDC CODE IS THE SUM OF ALL THE LOWER DDC CODES. FOR EXAMPLE, THE LINE 000 CONTAINS THE SUM OF THE NODES 000 TO 099 FOR EACH SEARCH ENGINE.

DDC	Google	Teoma	AOL	MSN	ASK	AV	Wisnut	Yahoo	Mean	Variance
000	0.815	0.820	0.861	0.730	0.820	0.743	0.895	0.700	0.798	0.004584
100	0.081	0.063	0.074	0.110	0.062	0.104	0.065	0.109	0.083	0.000436
200	0.023	0.021	0.020	0.022	0.021	0.025	0.020	0.027	0.022	0.000006
300	0.055	0.037	0.051	0.048	0.037	0.051	0.046	0.052	0.047	0.000049
400	0.335	0.386	0.295	0.452	0.386	0.395	0.038	0.429	0.340	0.017252
500	0.201	0.169	0.173	0.167	0.169	0.177	0.160	0.191	0.176	0.000187
600	0.192	0.133	0.170	0.160	0.133	0.174	0.174	0.186	0.165	0.000484
700	0.273	0.285	0.240	0.376	0.285	0.376	0.300	0.394	0.316	0.003290
800	0.169	0.145	0.143	0.160	0.145	0.185	0.129	0.194	0.159	0.000511
900	0.182	0.156	0.161	0.164	0.156	0.203	0.167	0.218	0.176	0.000546

TABLE XIII

NORMALISED RESULTS FROM HIGHEST CONFIDENCE AND SUPPORT QUERY PROBES. TO COMPARE SEARCH ENGINES OF DIFFERENT SIZES THE RESULTS ARE NORMALIZED USING A FROBENIUS NORM TO A FLOATING POINT VALUE BETWEEN ZERO AND ONE.

purpose search engines. Accuracy of results also depends on the quality of the training set and ontology. The purpose of this work was to present a new *method* for search engine analysis, with the intention of further refining the model in ongoing research and development.

We make a number of contributions to the field of ontologies, information retrieval(IR) and search engine analysis, the first being the creation of a large multi-domain ontology for representation of world knowledge for Web Intelligence. The second contribution is the evaluation of the search engines using both world knowledge and singular value decomposition. Finally a method of selecting query probe terms from the ontology is presented.

In further work the taxonomy will be extended to the fourth level of the Dewey Decimal system. We also need to implement improved ranking methods which consider part of speech terms used across different subjects. Further improvements in

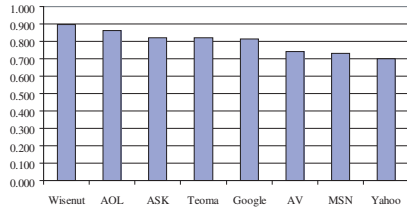
the size and quality of the terms selected will produce better and more accurate results. Also parsing the top k returned pages from each engine for each query probe and then using the data would make the method more powerful. We also intend to use this method to classify the large image search engines such as Google Images or MSN Images.

A. Acknowledgements

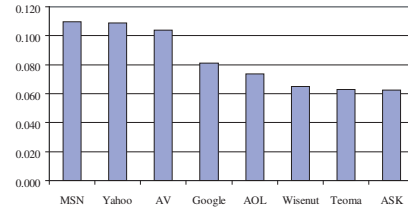
Thanks to Dr. Sylvia Edwards, Michael Gardner, Prof. Jiming Liu, Terry Hornby and Rae Westbury for reviewing this paper.

REFERENCES

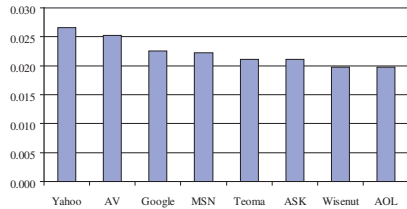
- [1] Steven M. Beitzel, Eric C. Jensen, Abdur Chowdhury, David Grossman and Ophir Frieder. Using manually-built web directories for automatic evaluation of known-item retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 373–374, New York, NY, USA, 2003. ACM Press.



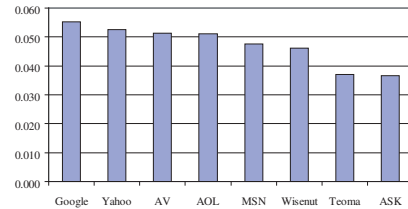
(a) 000 Generalities



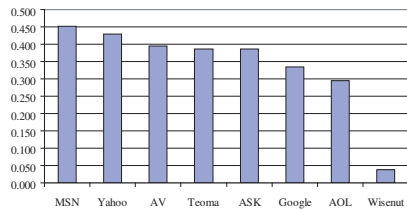
(b) 100 Philosophy & Psychology



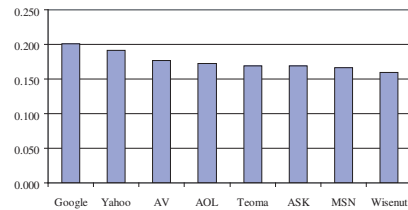
(c) 200 Religion



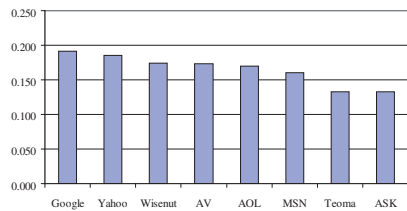
(d) 300 Social Sciences



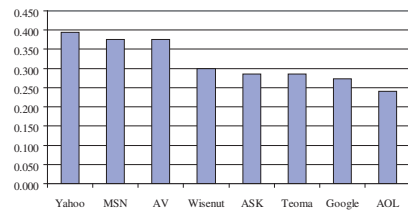
(e) 400 Language



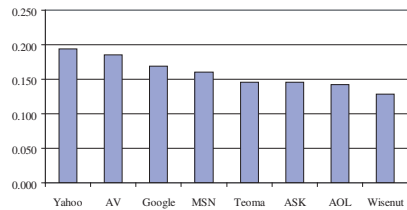
(f) 500 Natural sciences & mathematics



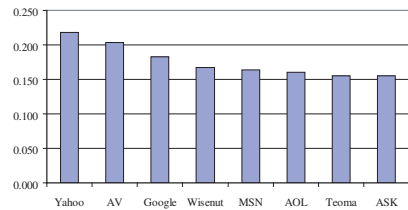
(g) 600 Technology (Applied sciences)



(h) 700 The Arts



(i) 800 Literature & rhetoric



(j) 900 Geography & history

Fig. 10. Normalised Rankings of Search Engine Content Distribution

- [2] P. Buitelaar. *CoreLex: Systematic Polysemy and Underspecification*. Ph.D. thesis, Computer Science Department, Brandeis University, 1998.
- [3] French J.C. Powell A.L. Callan, J. and M. Connell. The effects of query-based sampling on automatic database selection algorithms. In *Technical Report CMU-LTI-00-162*, Carnegie Mellon University, 2000. Language Technologies Institute, School of Computer Science.
- [4] J. P. Callan, Z. Lu and W. Bruce Croft. Searching Distributed Collections with Inference Networks. In E. A. Fox, P. Ingwersen and R. Fidel (editors), *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, Seattle, Washington, 1995. ACM Press.
- [5] Jamie Callan and Margaret Connell. Query-based sampling of text databases. *ACM Trans. Inf. Syst.*, Volume 19, Number 2, pages 97–130, 2001.
- [6] Jamie Callan, Margaret Connell and Aiqun Du. Automatic discovery of language models for text databases. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data*, pages 479–490. ACM Press, 1999.
- [7] Abdur Chowdhury and Ian Soboroff. Automatic evaluation of world wide web search services. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 421–422, New York, NY, USA, 2002. ACM Press.
- [8] Nicholas Eric Craswell. *Methods for Distributed Information Retrieval*. Ph.D. thesis, The Australian National University, 2001.
- [9] Nick Craswell, Peter Bailey and David Hawking. Server selection on the world wide web. In *Proceedings of the fifth ACM conference on Digital libraries, San Antonio, Texas, United States*, pages 37–46. ACM Press, 2000.
- [10] Daryl J. D'Souza, James A. Thom and Justin Zobel. A comparison of techniques for selecting text collections. In *Proceedings of the 11th Australasian Database Conference(ADC'2000)*, pages 28–32, Canberra, Australia, 2000.
- [11] F. Esposito, S. Ferelli, N. Fanizzi and G. Semeraro. Learning from parsed sentences with INTHELEX. In Claire Cardie, Walter Daelmans, Claire Nédellec and Erik Tjong Kim Sang (editors), *Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000*, pages 194–198. Association for Computational Linguistics, Somerset, New Jersey, 2000.
- [12] D. Faure and C. N'edellec. A corpus-based conceptual clustering method for verb frames and ontology acquisition. In *In LREC workshop on Adapting lexical and corpus resources to sublanguages and applications, Granada, Spain, 1998*.
- [13] James C. French, Allison L. Powell, James P. Callan, Charles L. Viles, Travis Emmitt, Kevin J. Prey and Yun Mou. Comparing the performance of database selection algorithms. In *Research and Development in Information Retrieval*, pages 238–245, 1999.
- [14] Fabien Gandon. Agents handling annotation distribution in a corporate semantic web. *Web Intelligence and Agent Systems, IOS Press*, Volume 1, Number 1, pages 23–46, 2003.
- [15] Michael Gordon and Praveen Pathak. Finding information on the world wide web: the retrieval effectiveness of search engines. *Inf. Process. Manage.*, Volume 35, Number 2, pages 141–180, 1999.
- [16] Luis Gravano and Héctor García-Molina. Generalizing GLOSS to vector-space databases and broker hierarchies. In *International Conference on Very Large Databases, VLDB*, pages 78–89, 1995.
- [17] Luis Gravano, Héctor García-Molina and Anthony Tomasic. GLOSS: text-source discovery over the Internet. *ACM Transactions on Database Systems*, Volume 24, Number 2, pages 229–264, 1999.
- [18] Luis Gravano, Panagiotis G. Ipeirotis and Mehran Sahami. Qprober: A system for automatic classification of hidden-web databases. *ACM Trans. Inf. Syst.*, Volume 21, Number 1, pages 1–41, 2003.
- [19] David Hawking, Nick Craswell, Peter Bailey and Kathleen Griffiths. Measuring search engine quality. *Information Retrieval*, Volume 4, Number 1, pages 33–59, 2001.
- [20] David Hawking, Nick Craswell and Kathleen Griffiths. Which search engine is best at finding online services? In *WWW Posters*, 2001.
- [21] David Hawking, Nick Craswell, Paul Thistlewaite and Donna Harman. Results and challenges in Web search evaluation. *Computer Networks (Amsterdam, Netherlands: 1999)*, Volume 31, Number 11–16, pages 1321–1330, 1999.
- [22] David Hawking and Paul Thistlewaite. Methods for information server selection. *ACM Transactions on Information Systems (TOIS)*, Volume 17, Number 1, pages 40–76, 1999.
- [23] David Hawking and Paul Thomas. Server selection methods in hybrid portal search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 75–82, New York, NY, USA, 2005. ACM Press.
- [24] P. Ipeirotis and L. Gravano. Distributed search over the hidden web: Hierarchical database sampling and selection. Technical report, Columbia University, Computer Science Department, 2002.
- [25] P. G. Ipeirotis, Luis Gravano and Mehran Sahami. Persival demo: categorizing hidden-web resources. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, page 454, New York, NY, USA, 2001. ACM Press.
- [26] Panagiotis G. Ipeirotis, Luis Gravano and Mehran Sahami. Automatic classification of text databases through query probing. In *Selected papers from the Third International Workshop WebDB 2000 on The World Wide Web and Databases*, pages 245–255, London, UK, 2001. Springer-Verlag.
- [27] Bernard J. Jansen, Amanda Spink and Tefko Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, Volume 36, Number 2, pages 207–227, 2000.
- [28] Joerg-Uwe Kietz, Alexander Maedche and Raphael Volz. A method for semi-automatic ontology acquisition from a corporate intranet. In *Proceedings of EKAW-2000 Workshop "Ontologies and Text", Juan-Les-Pins, France, October 2000*, number 1937 in Springer Lecture Notes in Artificial Intelligence (LNAI), 2000.
- [29] John D. King. Deep web collection selection. Master's thesis, School of Software Engineering, Queensland University of Technology, 2003.
- [30] John D King. Large scale analysis of search engine content. In *The Fourth International Conference on Active Media Technology, Brisbane, Australia*, Volume 1, page 451 to 453, 2006.
- [31] John D. King and Yuefeng Li. Web based collection selection using singular value decomposition. In *IEEE/WIC International Conference on Web Intelligence (WI'03)*, pages 104–110, Halifax, Canada, 2003.
- [32] Steve Lawrence and C. Lee Giles. Searching the World Wide Web. *Science*, Volume 280, Number 5360, pages 98–100, 1998.
- [33] H. Vernon Leighton and Jaideep Srivastava. First 20 precision among world wide web search services (search engines). *J. Am. Soc. Inf. Sci.*, Volume 50, Number 10, pages 870–881, 1999.
- [34] Y. Li and N. Zhong. Capturing evolving patterns for ontology-based web mining. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 256–263, Beijing, China, 2004.
- [35] Y. Li and N. Zhong. Mining ontology for automatically acquiring web user information needs. *IEEE Transactions on Knowledge and Data Engineering*, Volume 18, Number 4, pages 554–568, 2006.
- [36] King-Lup Liu, Clement T. Yu and Weiyi Meng. Discovering the representative of a search engine. In *CIKM*, pages 652–654, 2002.
- [37] Z. Lu, J.P. Callan and W.B. Croft. Applying inference networks to multiple collection searching. Technical Report TR96-42, University of Massachusetts at Amherst. Department of Computer Science, 1996.
- [38] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [39] A Maedche and S Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, Volume 16(2), pages 72–79, 2001.
- [40] Alexander Maedche and Steffen Staab. Discovering conceptual relations from text. In W. Horn (editor), *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*, pages 321–325, 2000.
- [41] Alexander Maedche and Steffen Staab. Learning ontologies for the semantic web. In *SemWeb*, 2001.
- [42] Weiyi Meng, King-Lup Liu, Clement T. Yu, Wensheng Wu and Naphtali Rische. Estimating the usefulness of search engines. *15th International Conference on Data Engineering (ICDE'99)*, Volume 1, pages 146–153, 1999.
- [43] Oxford University Press. Dictionary facts. <http://www.oed.com/about/facts.html>, 2005.
- [44] Gerard Salton and Chris Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.
- [45] L. Si and J. Callan. Relevant document distribution estimation method for resource selection, 2003.
- [46] Luo Si and Jamie Callan. Modeling search engine effectiveness for federated search. In *SIGIR*, pages 83–90, 2005.
- [47] A. Spink, D. Wolfram, B. Jansen and T. Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Science*, Volume 53(2), pages 226–234, 2001.
- [48] Nenad Stojanovic. Information-need driven query refinement. *Web Intelligence and Agent Systems, IOS Press*, Volume 3, Number 3, pages 155–170, 2005.

- [49] H. Suryanto and P. Compton. Learning classification taxonomies from a classification knowledge based system. In C. Nedellec P. Wiemer-Hastings S. Staab, A. Maedche (editor), *Proceedings of the Workshop on Ontology Learning, 14 Conference on Artificial Intelligence (ECAI'00)*, Berlin, 2000. Conference on Artificial Intelligence (ECAI'00).
- [50] Jason Chaffee Susan Gauch and Alexander Pretschner. Ontology-based personalized search and browsing. *Web Intelligence and Agent Systems, IOS Press*, Volume 1, Number 3, pages 219–234, 2003.
- [51] Xiaohui Tao, John D King and Yuefeng Li. Information fusion with subject-based information gathering method for intelligent multi-agent models. In *The Seventh International Conference on Information Integration and Web-Based Applications and Services, Kuala Lumpur, Malaysia*, Volume 2, page 861 to 869. iiWAS, 2005.
- [52] Roelof van Zwol and Herre van Oostendorp. Google's "i'm feeling lucky", truly a gamble? In *WISE*, pages 378–389, 2004.
- [53] C. Welty. Dls for dls : Description logics for digital libraries. In *Proc. of the 1998 Int. Workshop for Description Logics*, Trento, Italia, 1998.
- [54] C. Welty. The ontological nature of subject taxonomies. In *Proceedings of the 1998 International Conference on Formal Ontology in Information Systems (FOIS'98)*, 1998.
- [55] Ruchi Kalra Welty, Chris and Jennifer Chu-Carroll. Evaluating ontological analysis. In A. Halevy A. Doan and N. Noy (editors), *Proceedings of the ISWC-03 Workshop on Semantic Integration.*, 2003.
- [56] J. Xu and J. Callan. Effective retrieval with distributed collections. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 112–120, 1998.