An Integrating Text Retrieval Framework for Digital Ecosystems Paradigm

Dengya Zhu and Heinz Dreher

Curtin University of Technology, GPO Box U1987, 6845 Perth, Western Australia,
e-mail: dengya.zhu@postgrad.curtin.edu.au, h.dreher@curtin.edu.au

***Abstract*—The purpose of the research is to provide effective information retrieval services for digital 'organisms' in a digital ecosystem by leveraging the power of Web searching technology. A novel integrating digital ecosystem search framework (a new digital organism) is proposed which employs the Web search technology and traditional database searching techniques to provide economic organisms with comprehensive, dynamic, and organization-oriented information retrieval ranging from the Internet to personal (semantic) desktop.**

***Index Terms*—information retrieval, self-organizing search, categorization, crawler.**

I. Introduction

With the increasing emphasis on a digital ecosystem view of Information Systems worlds there is scope to create and adapt search functions to suit this new environment. To satisfy individual organisms, searches will need to be highly specialised both in query formulation and results presentation. There needs to be flexibility to connect to a variety of knowledge repositories, consult knowledge structuring schemas in the form of ontologies, thesauri, taxonomies and the like, and the ability to construct and interact with organism-specific and organisation-oriented ontological filters which act to sieve out relevant items and to pre-select relevant candidate items. As an organism searches it learns how to adapt to the environment with which it is interacting both to facilitate the current search but also to inform future search activity. A truly digital ecosystem-aware search system would provide for learning such self-organising and accommodating behaviour.

To survive in the digital ecosystem, an organization should adapt the digitalization trends, and strive to leverage their data for competitive advantage [5]. Within this ecosystem, information publication on the Web is for an organization akin to a person making him/herself known to others via personal presentations; and information retrieval for an organization is akin to a person learning and acquiring knowledge about the ecosystem environment upon which survival depends. In this scenario, information processes and services are at the core of digital ecosystem.

[5] state that there are three existing toolkits for enterprise searchers. The first type is *raw search engines*, such as the Open Source Apache project Lucene [14][19] which handles Boolean logic, fuzzy queries, stemming, indexing, and hit highlighting. Commercial providers may add an entity extractor, thesaurus, automated classification and other key features. The second type is *intranet appliances* which are low-customization boxes that can simply plug into a network, point to a data source, and compile indexes. The enterprise search products of Google and Thunderstone are available now and reviews are positive [1][12][13][29]. *Desktop search* is the third type searching tool that focuses on searching of e-mails, local files, instant messaging history, Web history, contacts and more. It is suggested that to solve enterprise-scale problems, a searching system should combine tagging, categorization, and navigation to improve the overall experience for end users. An enterprise metadata category – an ontology used to manage metadata – can be built as follows: 1) define a metadata schema, 2) index a set of documents, and 3) write a user interface for querying and displaying results. While automatic metadata extraction is never perfect, a user interface is needed to allow modification and re-use of the metadata that was found. An integral system should also satisfy scalability, security, metadata update, view privilege, and query optimization criteria.

II. The Search Experience

In a digital ecosystem environment, one can imagine search organisms interacting with knowledge repositories, retrieval interfaces for composing and refining search queries. In their Query-formulate/Find/Reformulate table [9] suggest a series of steps which accomplishes just this and concludes with results presentation. The current experience in information retrieval is far from satisfying [36]. The following three issues are challenging information retrieval in a digital ecosystem.

A. Existing search tools are not integrated

In addition to the general purpose Web search engines such as Google, Yahoo!, MSN, there are also specific search tool functions which are desirable, for example, desktop search, music search, language specific search, and specific full text database and bibliographic searching. While the specific search tools provide more effective search for a specific domain or field as compared with the general purpose search engines, information seekers must install these tools on their computers, and then match the search tool/function with the information retrieval need. This process may involve considerable trial and error and investment in learning the specific methods of a variety of systems. For example, when searching for a full text academic paper the researcher may first try Google. If Google does not provide full text of the paper, the searcher may try a

specific full text database such as the ACM digital library, or a specific e-journal. Jumping from one search engine to another search tool and again to another tool is time consuming, disorientating, and can be discouraging. An integral search framework which combines all of the search tools can facilitate this situation.

B. Search results are not very well organized

Another problem is the so called information overload [36]. General purpose search engines usually returns thousands, even millions, of search results in a plain list format which are ranked according to the syntactic similarities between query and indexed documents (Web pages) [3]. Although some search engines, such as Clusty.com (www.clusty.com) attempt to cluster the search results, the clustering algorithms are far from perfect. In digital ecosystem environments users naturally adapt to feedback: initial search results will generate interaction from users [9] to prompt another search based on the current search-query/search-result set refined by some further constraint information such as categorization, similarity modelling, and ontological filtering [9].

C. Search results are not personalized

Most search engines and tools try to return and rank search results suitable for general purpose search; personalized search is not considered [11]. No matter what role a searcher has - a car salesman, an environmentalist, or a computer technician - if they use the same query "jaguar", they will get exactly the same search results. The salesman may be concerned only about the jaguar car; the environmentalist is seeking information about the animal; and the technician is thinking of using the Apple's Jaguar operating system. The quality of search results of general purpose search tools consequently needs to improve.

To address the above problems we propose an integrating text retrieval framework.

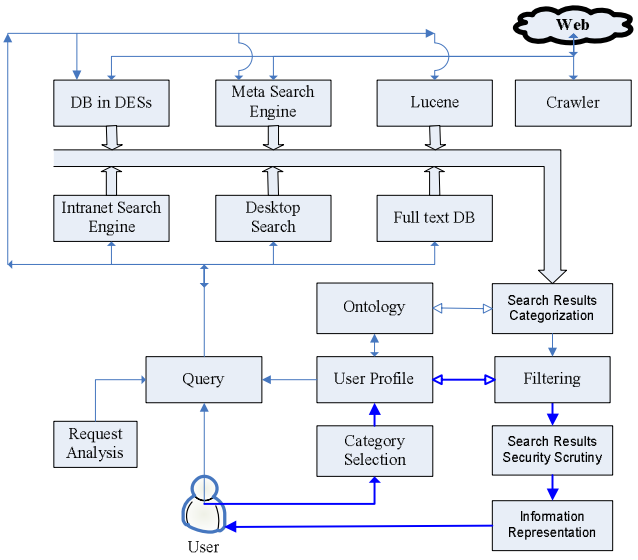

Fig. 1 A search framework in digital ecosystem

III. AN INTEGRATING DIGITAL ECOSYSTEM SEARCH FRAMEWORK

A novel integrating digital ecosystem search framework is proposed which leverages the power of Web search technology and traditional database and text repository searching techniques to provide organizations with comprehensive, dynamic, and organization-oriented information retrieval ranging from the Internet to personal desktop.

Fig 1 illustrates the proposed digital ecosystem search framework.

Components in this framework are described in the following sections.

A. Crawler

"A crawler is a program that downloads and stores Web-pages, often for a Web search engine." [7] Alternative terms used for *crawler* are *robot*, *spider*, *worm*, *bot*, and *ant*. Many challenges are facing crawlers due to the huge size, complexity and rapid growth of the Web - page selection, importance, recency, and refresh issues. In addition to regular page refresh, crawlers in digital ecosystems should concentrate on the effectiveness of the crawl. Generally, a crawler starts off with an initial set of URLs which are placed in a queue where they may be prioritized. A URL is selected based on some ordering strategy, the crawler downloads the page, extracts URLs in the downloaded page, and puts the new URLs in the queue. This process is repeated until the crawler decides to stop [3] [7]. Prioritizing the URLs in the queue and setting the stop conditions are both related to estimating, or measuring the **relevance** of the URL content to the semantic need of the digital ecosystem. However, the relevance *per se* is an arguable topic [21][36].

One solution to this issue is divided into four stages as follows. The first stage is to keep an active URLs list which contains all necessary relevant Web-sites as determined by digital ecosystem users. This list is dynamic for it can be updated when a newly relevant URL is found.

The second stage is to use the list to initialize the crawling queue, download relevant Web-pages, and consequently build an initial relevant document repository. After downloading all the Web-pages in the list, URLs are extracted from the Web-pages. The Web-pages pointed to by the extracted URLs are also downloaded (second level Web-pages). Both the first level downloaded and second level downloaded Web-pages are the source of the initial document repository.

The third stage is to build the initial relevant document repository. Each downloaded Web-page is treated as a document, indexed, and stored in the retrieval framework.

The last stage is to index and filter the crawled Web-pages. The URLs are extracted from the second level Web-pages downloaded at stage 2. All the extracted URLs are formed into a new queue, and crawling is conducted based on this new queue with all Web-pages being downloaded from the URLs. However, it is highly possible not all the downloaded Web-pages at this stage are relevant to the searcher. For example, a mining corporation in Western Australia may have a link to Curtin University because graduates from Curtin University are working there. Some

Web-pages of Curtin University are also related to the mining industry; however, it is obvious that not all Web-pages of Curtin University are about mining industry. Therefore, each downloaded Web-page is filtered based on the relevance of the Web-page to the searcher.

Because the initial document repository is believed relevant to the searcher, it can be used to filter the new crawled Web-pages. As pointed out by [6], information filtering and information retrieval are actually two sides of the same coin. Several existing information modelling methods, such the Vector Space Model [30], Probabilistic Model [16], and text classification techniques [31] can be used to filter the crawled Web-pages.

B. Databases in Repurposing

Existing database in an enterprise are usually managed by a Relational Database Management System (RDBMS), this type of application can be plugged into the search framework. Extracted metadata can also be managed by the RDBMS. Security sensitive data are also managed by the RDBMS which provides also security and audition management [32].

[5] suggest custom database information can be repurposed. For example, for a database holding metadata about thousands of multimedia files, an XML file configured database crawler can map columns (such as language, series, business unit, date, speakers, etc) in a table to values in an ontology by using ontology development tools such as Protégé (http://protege.stanford.edu/).

C. Desktop search

As pointed out by [5], Google, Microsoft, Yahoo!, and other major players provide free downloaded desktop search solutions which enable searching of anything on one's computer almost instantaneously – as fast as one can type. As the data stored in personal computers increases to hundreds of gigabytes and beyond, desktop search will no doubt greatly improve the productivity of users in digital ecosystem environments. Powerful search tools providing relevant search-results will reduce the need for users to manually construct and maintain models of the knowledge they store on their desktop repositories. In place of these models acting as indexes to information, one will have dynamic integrated retrieval systems to act as an agent to return pointers or links to the desired information.

D. Full text DataBase Retrieval

Full text database retrieval differs from database searching in which exact matching is necessary. Database retrieval languages (such as SQL) facilitate composing queries to find all objects which match clearly defined conditions like those defined in an algebraic expression; any mismatching among thousands of objects is thus an error. However, in a full text database retrieval system there is more tolerance for error. This is mainly because Information Retrieval (IR) usually deals with natural language text which is not always well structured, and could be semantically ambiguous [4].

Another difference between full text database retrieval and data retrieval is that a full text retrieval system is always trying to retrieve information about a subject or topic. To be effective in trying to satisfy the user information need, the full text retrieval system needs to 'interpret' the contents of the text objects in a collection and rank them according to a degree of relevance to the search term. This process of 'interpretation' involves extracting both syntactic and semantic aspects from the text objects, and using the extracted information to match the user information need. A full text database retrieval system concerns not only the syntactic interpretation of search terms and text objects, but also the relevance of an object to the user information need [4].

Both open source search engines, such as Lucene [19] and Xapian [33], and commercial systems such as Northern Light [24] are available for full text search. Lucene is seeing increasing use by third-party applications and we think it is appropriate to explore the use of Lucene in this digital ecosystem search framework.

E. Information Representation

In the integrating digital ecosystems search framework, search results are obtained from traditional database search, full text database, intranet, Internet, and desktop searches - all results being integrated into one coherent information representation.

Users of the integral DES search framework should be able to choose which data sources are to be used in the retrieval. One user may only be interested in search results from their Management Information System (MIS) system, another may need Web information from meta-search engines, while a third user may retrieve information from an intranet, and yet another user may need information from both intranet and Internet. The integrating digital ecosystem search framework needs to permit users to set the search scope and thus provide the flexibility to access data sources to satisfy their needs.

To deal with the huge number of search results returned from meta-search engines or intranet search engine, results categorization based on a domain ontology is one approach to meaningfully represent the search results [36]. Some ([17] [23]) use *Yahoo! Directory* as a lightweight ontology to classify search results. Northern Light search engine [24] provides *Custom Folders* to organize search results. The folders are automatically created according to the four dimensions: subject; source; type; and language. The first one is *subject folders* that use a hierarchy of over 200,000 subject terms created by the librarians on Northern Light's staff. Northern Light uses page based word occurrences, matching the occurrences of keywords to the subject dimension. It does not reliably identify the subject of all Web-pages, but rather is a rough approximation. *Source folders* can be one specific publication. This dimension is only available for search results from the special collections database (e.g., commercial sites, personal Web-pages, magazines, encyclopaedias, databases). The third dimension is the *type folders*. Examples of this kind of folder are press releases, product, reviews, and resumes. The last dimension is *language folders*. There is no global information provided about the category structure or about the distribution of search results across categories [2] [10] [25]. Each folder is a one or

two word label, and documents that contain the label are organized under the label. Northern Light does not reveal the approach used to create the folders [35].

To create an easily understood, concise knowledge skeleton, we have found that *Yahoo! Directory* or Open Directory Project (ODP) [26] may be too broad to suit the specific digital ecosystem. [5] suggest using the ontology facets (role restrictions of the properties of an ontology) to classify search results. In this research, however, a light-weight ontology which combines the *Yahoo! Directory* or the *ODP* with the user-organization ontology is proposed, and each of the categories in this lightweight ontology should also have a description facet which manifests the semantic interpretation of the category, and according to which the search results can thus be classified automatically. When a specific category is selected by a user, the search results will be filtered based on the selected category [36]. An example user interface is shown in Fig 2.

F. Intranet Search Engine

Information now accessible in most intranets is increasing dramatically, and searching information in the intranet is somewhat of a daunting task. This issue has been addressed by search tool development companies such as Google and

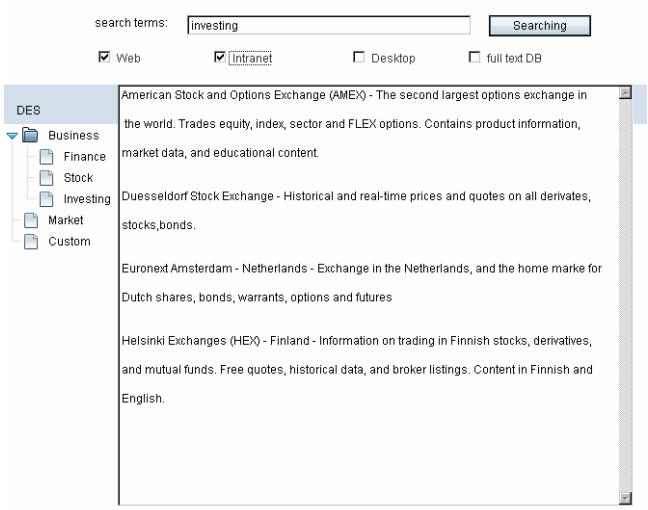

Fig. 2 A sample user interface in digital ecosystem search framework

Thunderstone. [5] indicate intranet search tools are low-customization boxes that one can simply plug into an intranet, point to a data source, and let indexing proceed in background mode. However, depending on the ranking algorithms used by the vendor, the performance of these Intranet Search Engines differs dramatically. ISYS Search Software [15] suggest before deployment of an intranet search engine, issues such as objectives, data source and file type, technology environment, features, installation and maintenance, pricing structure and vendor credentials should be considered carefully.

G. Lucene

Lucene is a cross-platform, high-performance, full-featured text database search engine [14] which can be employed in the integrating search framework of digital ecosystems. Lucene is a free text indexing and searching API written in Java, and ports are also available in Perl, Python, C++, MS .Net, Ruby, and so on. It is a member of the Apache Jakarta family (http://jakarta.apache.org/) of projects, licensed under the liberal Apache Software License (http://www.apache.org/foundation/licence-FAQ.html). Lucene is composed of two main independent parts: text indexing; and text searching, although indexing inherently affects searching outcomes [14].

H. Meta-search engine

The digital ecosystem search framework should also include a meta-search engine. [20] define meta-search engine as "a system that provides unified access to multiple existing search engines." [4] and [20] point out that the introduction of meta-search engine is mainly based on the reasons of: single search engine's processing power may not scale to the tremendous increase and virtually unlimited amount of data; it is difficult or even impossible for a single search engine to gather all the data on the Web and keep it up to date; and some "deep web" may not allow their documents to be crawled by external websites, but allow their documents to be accessed by their search engine only. These reasons are also applicable to our need here, the integrating search framework suitable for digital ecosystem users.

Fig. 3 illustrates a conceptual architecture of a meta-search engine. Users' queries are first analyzed and a set of suitable databases (coupled with search engines) are selected by the *database selector*. *Document selector* decides either the number of documents that should be retrieved from the component search engines, or a local simi-

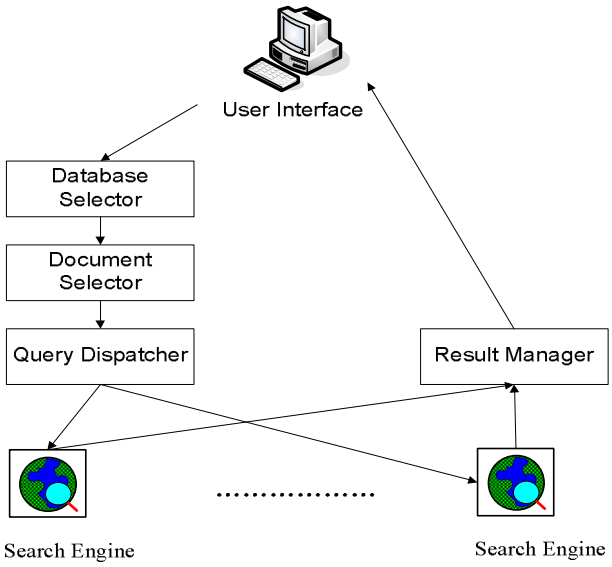

Fig. 3 Meta-search engine structure[20]

larity threshold is used to limit the documents retrieved from the component search engine. *Query dispatcher* establishes a connection with the server of each selected search engine and passes the query to it. The returned search results from selected component search engines are merged by *results manager,* which combines all the results into a single ranked list and renders it to the user [20]; in this research, the *Search Results Categorization* component (Fig. 1).

I. Lightweight ontology

[5] propose an enterprise metadata catalogue to benefit business applications. A metadata schema is first defined which may include the date, author, subject, and keywords of documents and so on. Metadata can be extracted from documents by using tools such as entity extractor which extract entity names. It can also be extracted from custom database and applications such as Microsoft SharePoint.

To improve search effectiveness, a lightweight ontology as mentioned in section *E. Information Representation* above, is also needed in the digital ecosystem search framework. This lightweight ontology serves as an hierarchical knowledge structure according to which search results can be categorized [36]. Based on user selection, search results are also filtered and only search results classified under this category are presented to the user. Our recent initial trials have demonstrated the use of lightweight ontology to categorize and filter search results can improve performance of search engines - more than 23% precision improvement can be obtained [36].

J. Personalization and user profile

Personalized information retrieval concerns not only retrieving syntactically relevant information, but also a user's information consumption pattern, searching strategies, application used and the nature of the information, as indicated by [28]. User profile is the core in personalization searching, although there is no agreement on what a user profile should include at the present. [11] indicate a user profile is a reference ontology in which each concept has a weight indicating the user's interest in that concept. [28] use information space of the ODP to represent the user model. [8] propose a user model which combines two proposed standard learner profiles: IEEE Personal and Private Information (PAPI) [27]; and IMS Learner Information Project (LIP)[18]; to express the features of a user. In the digital ecosystem search framework, the latter user profile is preferred as this emphasises the facilitation of information flow between computational systems.

K. Query and request analysis

According to [4], query is "the expression of the user information need in the input language provided by the information system. The most common type of input language simply allows the specification of keywords and of a few Boolean connectives." This means that a query comes from a user's *information needs*. As stated by [20][22], when a user has a problem, or an aim to achieve, the users is in a "problematic situation" that needs information for resolution or solution. The user perceives the problem and builds a mental, *Information Need*, to implicitly represent the problematic situation. The user then expresses the information need in a *request*, an expression of the information need in a human language, usually in natural language. The request must be translated into a *query*, a form understandable by an information retrieval system in a "system" language, such as that based on Boolean logic or in an iterative process as suggested by [9].

As stated by [28], query augmentation and result processing are two primary usage of user profile. In the DES search framework, when a user submits a query, the information in the user profile is used to refine or add other terms to the submitted query by comparing the query with the contextual information in the user profile.

L. Filtering

Another usage of the user profile is search results filtering. Based on the pre-built user profile, search results are compared with the features in the profile; they are re-ranked [11] and then only search results having similar features described by the user profile are presented to the user [36]. Other search results will be filtered out. Traditional IR model, such as VSM and probabilistic model can be used for this purpose. Text categorization techniques [31] can also serve this intent.

M. Search results categorization

Text categorization is the problem of automatically assigning predefined categories to free text documents [34]. [23] uses *Yahoo! Directory* as an automatic Web-page classifier, for each of the top level *Yahoo! Directory*, a separate Naive Bayesian classifier being constructed and trained for both positive and negative examples. [17] use tf-idf weighting scheme and probabilistic retrieval model to classify web documents under the hierarchical structure of *Yahoo! Directory*. A comprehensive review of machine learning technology in text classification is presented by [31].

In digital ecosystems environments, search results are categorized based on the lightweight ontology as discussed previously. Categorization techniques are available as discussed by [31].

N. Search results security scrutiny

This component performs a search results security scrutiny task which concerns "who is allowed to update a piece of metadata and who is allowed to view a particular piece of metadata about a document (or know that the document exists at all)" [5] .

IV. ADAPTATION AND SELF ORGANISING BEHAVIOUR

In a digital ecosystem environment, adaptation and feedback are a natural part of all transactions. User search behaviour is thus interactive and permits stepwise or incremental refinement of the search query [9]. Refining the query in parallel with results categorization and subsequent filtering, helps assure superior search outcomes. The actions and decisions made during an interactive search session, for example, the category selection action, provide input to user profile construction and specialised ontology development and can be used in future search scenarios to both speed up the finding of relevant documents and limit the scope of search target repositories. Such self-organising behaviour is characteristic of the digital ecosystems paradigm.

V. FUTURE WORK

Part of the framework has been constructed and trialed

and the results of experiments are encouraging – an average improvement of more than 23% precision is achieved over one comparable system [36]. Still under development are the intranet search, user profile creation, search results security scrutiny, and the assembling of all the search components into one framework. As information retrieval technology is also developing rapidly, some components in the framework may be modified, some new components may be added while others may be removed.

VI. SUMMARY

In this paper, a novel search framework aiming at providing effective information retrieval services for digital organisms in digital ecosystem environments is proposed. The search framework integrates not only traditional database search (MIS) and Web search (search engine), but also intranet search, desktop search, full text database search, personalization, ontological search results categorization and search results security scrutiny. Experiments on some of the search components so far have demonstrated improvements in the significance of the search results. Further development work is needed to complete the whole framework and conduct evaluation studies.

VII. REFERENCES

[1] E.N. Albro, "Google Mini is a Mighty Search Tool", PC World, June 21, 2006, http://www.pcworld.com/article/id,126139-page,1/article.html

[2] J. Allen and H. Raghavan, "Using Part-of-speech Pattern to Reduce Query Ambiguity", in *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval (SIGIR-02)*, ACM Press, New York, NY, pp. 307-314.

[3] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke and S. Raghavan, "Searching the Web", *ACM Transactions on Internet Technology*, 2001, vol. 1, no. 1, pp. 2-43.

[4] R. Baeza-Yates and B. Ribeiro-Neto, 1999, *Modern Information Retrieval*, ACM Press, New York & Addison Wesley, Harlow.

[5] R. Barrows and J. Traverso, "Search Considered Integral", *ACM QUEUE*, May 2006, 30-36.

[6] N.J. Belkin and W.B. Croft, "Information Filtering and Information Retrieval: Two Sides of the Same Coin", *Communication of the ACM,* Nov. 1992, vol. 35, no. 12, 29-38.

[7] J. Cho and H. Garcia-Molina, "Parallel Crawlers", in *Proceedings of the eleventh international conference on World Wide Web,* ACM Press, New York, NY, 2002, 124-135.

[8] P. Dolog and W. Nejdl, "Challenges and Benefits of the Semantic Web for User Modelling", in *Proceedings of the 12th International World Wide Web Conference*, 2003, 99-111.

[9] H. Dreher and B. Williams, "Assisted Query Formulation Using Normalised Word Vector and Dynamic Ontological Filtering", in *Proceedings of the 7th International Conference on Flexible Query Answering Systems H. Legind Larsen et al. (Eds.) (FQAS 2006)*, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 282-294.

[10] S. Dumais, E. Cutrell and H. Chen, "Optimizing search by showing results in context", in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (SIGCHI'01)*, ACM Press, New York, NY, 277-284.

[11] S. Gauch, J. Chaffee and A. Pretschner, "Ontology-based personalized search and browsing". *Web intelligence and Agent System*, 2003, vol. 1, no. 3-4, 219-234

[12] R. Gedda, "Google enters local enterprise market", *ComputerWorld*, June 22 06 2006, http://www.computerworld.com.au/index.php/id;697672342;fp;16;fpid;0

[13] Google Enterprise Solution, http://www.google.com/enterprise/news_events.html

[14] O. Gospodnetic and E. Hatcher, *Lucene IN ACTION: A guide to the Java search engine*, Manning Publications Co., Greenwich, 2005.

[15] ISYS Search Software, "Selecting an Intranet Search Engine", 2004. Retrieved 2nd October, 2006, from http://www.isys-search.com/downloads/whitepapers/

[16] K.S. Jones, S. Walker and S.E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments", *Information Processing and Management*, 2000, vol. 36, no. 6, 779-840.

[17] C. Klas and N. Fuhr, "A new Effective Approach for Categorizing Web Documents", in *Proceedings of the 22nd Annual Colloquium of the British Computer Society Information Retrieval Specialist Group (BCSIGSG-00)*, 2000, Cambridge, UK.

[18] LIP, IMS Learner Information Project, http://www.imsglobal.org/

[19] Lucene, Apache Jakarta Project, http://lucene.apache.org/java/docs/

[20] W. Meng, C. Yu and K. Liu, "Building Efficient and Effective Metasearch Engines", *ACM Computing Surveys*, 2002, vol. 34, no. 1, 48-89.

[21] S. Mizzaro, "Relevance: The Whole (hi)story", *Journal of the American Society for Information Science*, 1997, vol. 48, no. 9, 810-832.

[22] S. Mizzaro, "How many relevances in information retrieval", *Interacting with Computers*, June, 1998, vol. 10, no. 3, 303-320.

[23] D. Mladenic, "Turning Yahoo into an Automatic Web-Page Classifier", in *Proceedings of the 13th European Conference on Artificial Intelligence Yong Research Paper*, 1998, John Wiley & Sons, Ltd, 473-474.

[24] Northern Light, http://www.northernlight.com/

[25] G. Notess, "Northern Light: New Search Engine for the Web and Full-Text Articles", *Database Magazine*, Feb-March, 1998, vol. 21, no. 1, 32-37.

[26] The Open Directory Project, http://www.dmoz.org

[27] PAPI, IEEE Personal and Private Information, http://edutool.com/papi/papi_learner_07_main.pdf#search=%22IEEE%20P1484.2%22

[28] J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar and T. Breuel, "Personalized Search: A contextual computing approach may prove a breakthrough in personalized search efficiency", *Communications of the ACM,* September 2002, vol. 45, no. 9, 50-55.

[29] T. Powell, "Thunderstone chases Google's search watermark", *Network World*, October, 31, 2005, http://www.networkworld.com/cgi-bin/mailto/x.cgi

[30] G. Salton, A. Wong and C.S. Yang, "A Vector Space Model for Automatic Indexing": *Commmunication of the ACM*, Nov. 1975, vol. 18, no. 11, 613-620.

[31] F. Sebastiani, "Machine Learning in Automated Text Categorization", *ACM Computing Surveys*, 2002, vol. 34, no. 1, 1-47.

[32] A. Silberschatz, H.F. Korth and S. Sudarshan, *Database System Concepts*, 3rd edt., McGraw-Hill, 1993, 633-644.

[33] The Xapian Project, http://www.xapian.org/

[34] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization", *Information Retrieval*, 1999, vol. 1, no. 2, 69-90.

[35] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration", in *Proceedings of the 19th ACM International Conference on Research and Development in Information Retrieval (SIGIR-98)*, ACM Press, New York, NY, 1998, 45-54.

[36] D. Zhu, "Improving the Relevance of Search Results via Search-term Disambiguation and Ontological Filtering", Master Thesis, Curtin University of Technology. To be submitted in January, 2007.