A Parallel Computing Approach to Creating Engineering Concept Spaces for Semantic Retrieval: The Illinois Digital Library Initiative Project

Hsinchun Chen, Bruce Schatz, Tobun Ng, Joanne Martinez, Amy Kirchhoff, Chienting Lin

Abstract:

This research presents preliminary results generated from the semantic retrieval research component of the Illinois Digital Library Initiative (DLI) project. Using a variation of the automatic thesaurus generation techniques, to which we refer as the *concept space* approach, we aimed to create graphs of domain-specific concepts (terms) and their weighted co-occurrence relationships for all major engineering domains. Merging these concept spaces and providing traversal paths across different concept spaces could potentially help alleviate the *vocabulary (difference) problem* evident in large-scale information retrieval. We have experimented previously with such a technique for a smaller molecular biology domain (Worm Community System, with 10+ MBs of document collection) with encouraging results.

In order to address the scalability issue related to large-scale information retrieval and analysis for the current Illinois DLI project, we recently conducted experiments using the concept space approach on parallel supercomputers. Our test collection included 2+ GBs of computer science and electrical engineering abstracts extracted from the INSPEC database. The concept space approach called for extensive textual and statistical analysis (a form of knowledge discovery) based on *automatic indexing* and *co-occurrence analysis* algorithms, both previously tested in the biology domain. Initial testing results using a 512-node CM-5 and a 16-processor SGI Power Challenge were promising. Power Challenge was later selected to create a comprehensive computer engineering concept space of about 270,000 terms and 4,000,000+ links using 24.5 hours of CPU time. Our system evaluation involving 12 knowledgeable subjects revealed that the automatically-created computer engineering

concept space generated significantly higher *concept recall* than the human-generated INSPEC computer engineering thesaurus. However, the INSPEC was more precise than the automatic concept space. Our current work mainly involves creating concept spaces for other major engineering domains and developing robust graph matching and traversal algorithms for cross-domain, concept-based retrieval. Future work also will include generating individualized concept spaces for assisting user-specific concept-based information retrieval.

index terms: semantic retrieval, concept space, concept association, parallel computing, digital library

1 Introduction

The Illinois Digital Library Initiative (DLI) project entitled: "Building the Interspace: Digital Library Infrastructure for a University Engineering Community," is one of six projects funded recently by NSF/ARPA/NASA. The goal of the project is to evolve the Internet into the *Interspace*, in particular by bringing professional and "intelligent" search and display of structured documents to the Net. To accomplish this, we are constructing a large-scale digital library testbed of SGML journal articles while concurrently performing the underlying research to provide effective interaction with such a library across networks. That is, we are critically concerned with both the functionality needed to interact with structured documents and the infrastructure required to scale such functionality up to a global community [36].

The remaining part of the paper presents in detail a parallel computing approach to creating engineering concept spaces for semantic retrieval in the Illinois DLI project.

2 Alleviating the Vocabulary Problem Using Concept Spaces

The *vocabulary (difference) problem* in human-computer interactions has been studied extensively in recent years [22]. In [22], Furnas et al. found that in spontaneous word choice for objects in five domains, two people favored the same term with less than 20% probability. This fundamental property of language limits the success of various design methodologies for vocabulary-driven interaction. In information science, indexing and search uncertainty have been recognized as the primary sources of information retrieval problems. Previous research [4] has shown that different indexers, well trained in an indexing scheme, might assign index terms for a given document differently. It has also been observed that an indexer might use different terms for the same document at different times (possibly because of learning or the cognitive state of mind at indexing). A high degree of uncertainty with regard to search terms has also been reported: searchers tend to use different terms for the same information sought. Because of the indeterminism involved in indexing and searching, an exact match between the searcher's terms and those of the indexer is unlikely [8]. This often results in poor recall and precision in search.

The vocabulary problem affects every domain of human knowledge. Based on research over the past few decades, it has become clear to information scientists that development of effective online information retrieval systems must consider the cognitive processes and the vocabulary association characteristics of the users.

2.1 Vocabulary Association and Thesaurus Work

According to Belkin, users of information retrieval systems bring with them a problem statement which represents an information need. Inherent in all information needs are "anomalous

states of knowledge" (ASKs) [5]. In Belkin's document retrieval system based on ASKs [5], the searcher's state of knowledge is represented as a network of associations between words. ¿From the structure and characteristics of the network, it is possible to identify anomalies in the state of knowledge. Several models of human memory association have been suggested wherein knowledge is represented by network-like structures with linked propositions. Anderson's work in human memory is particularly pertinent to term associations in retrieval [3]. According to Anderson, people remember not the exact wording of verbal communication, but the meaning underlying it. The smallest unit of knowledge that can stand as an assertion bearing meaning is the proposition. Memory, then, is represented as a network of such propositions. The strength of the association paths leading to that piece of information contributes to the level of activation being spread. This theory of *spreading activation* has influenced the design of many semantic network based information retrieval systems [13].

Many research groups have created vocabulary-based search aids for online information retrieval systems by making use of existing thesauri or dictionaries. Thesauri, in particular, exhibit a structure similar to human word-association networks. While these tools are able to provide the searcher with alternate terms to use in searching, they do not overcome the *knowledge acquisition bottleneck*: the cognitive demand required of humans (indexers or domain experts) to create thesauri or dictionaries in the first place. An alternative approach to creating vocabulary-based search aids is based on *automatic thesaurus generation*.

- **Incorporating existing thesauri:**

  Fox et al. focused on creation of so-called "relational thesauri." For example, CODER adopted the *Handbook of Artificial Intelligence* and *Collin's Dictionary* [20]. Ahlswede and Evens parsed [1] *Webster's Seventh New Collegiate Dictionary* to obtain a "lexical database" containing lexical or lexical-semantic relationships from the dictionary definitions. Lesk converted an online version of Murray's *Oxford English Dictionary* into a

thesaurus-like tool to facilitate searching of historical manuscripts. These approaches represent attempts to produce "universal lexicons," rather than domain-specific thesauri or dictionaries. Chen et al. conducted a series of experiments which included several large-scale, domain-specific thesauri. In [9], Chen and Dhar incorporated a portion of the *Library of Congress Subject Headings* (LCSH) in the computing area into a system that used a branch-and-bound spreading activation algorithm to assist users in query formulation. More recently, they developed concept-based document retrieval using multiple thesauri: two existing thesauri (LCSH and the ACM Computing Review Classification System) and an automatically-generated computing-specific thesaurus [11] [13]. The National Library of Medicine's *Unified Medical Language System (UMLS)* project is probably the largest-scale effort adopting existing domain-specific knowledge sources or thesauri in information access. It aims to build an intelligent automated system that understands biomedical terms and their interrelationships and uses this understanding to help users retrieve and organize information from multiple online sources [27].

- **Automatic thesaurus generation:**

  Numerous investigators have developed algorithmic approaches to *automatic thesaurus generation*. Most of these approaches employ techniques that compute coefficients of "relatedness" between terms using statistical co-occurrence algorithms (e.g., cosine, Jaccard, Dice similarity functions) [10] [34] [32]. Some algorithms, however, perform cluster analysis to further group terms of similar meanings [32]. Other algorithms, such as latent semantic indexing [19], perform statistical analysis to identify important semantic descriptors. Stiles [38] was one of the early researchers who reported improved retrieval performance using a method based on term association (with collections of librarian-applied subject tags). Doyle [18] further argued that the principles underlying

association-based retrieval should apply whether the associations are determined by humans or by machines (programs).

More recently, Crouch and Yang [17] automatically generated thesaurus classes from text keywords, which can subsequently be used to index documents and queries. Crouch's approach is based on Salton's vector space model and the term discrimination theory.

2.2 The Concept Space Principles

After closely examining previous research (both in information science and cognitive studies) and based on our own experience in creating domain-specific thesauri in several scientific, engineering, and business domains, we believe that creating robust and useful domain-specific thesauri (not universal thesauri) automatically requires a clear understanding of the following system development principles: *logarithmic vocabulary growth, completeness, term specificity, asymmetric association, relevance feedback, vocabulary overlapping, and spreading activation.* Many of these principles were developed based on human information processing theory [6] and our own information retrieval cognitive studies [8]. We refer to our approach to automatic thesaurus generation as a *concept space* approach because our goal is to create meaningful and understandable domain-specific *concept spaces* (networks of terms and weighted associations) which could represent the concepts (terms) and their associations for the underlying *information spaces* (i.e., documents in different domain-specific databases) and could assist in concept-based, cross-domain information retrieval. We review these principles below in the context of our research:

- **Logarithmic vocabulary growth principle:**

The most important rationale behind automatic thesaurus generation is related to the *information overload* problem. Lancaster has shown that the rate of growth for information (i.e. documents) continues at an exponential pace, while the corresponding rate of growth over the same period of time for number of concepts (keywords and terms) converges logarithmically [25]. This principle appears to be applicable to different scientific and engineering domains and is particularly relevant in light of the rapid proliferation of Internet servers and distributed databases.

- **Completeness principle:**

  Early information science researchers such as Lesk have suggested the importance of large document collections for generating automatic thesaurus [26]. This is especially true when considering the logarithmic vocabulary growth principle described above. If a collection which is used to generate an automatic thesaurus is limited, it is impossible to reach the plateau (or convergence) on the logarithmic curve. Many previous automatic thesaurus generation studies which used only selected collections and/or samples of documents in a subject domain suffered from a lack of completeness in document collections.

- **Term specificity principle:**

  Most prior automatic thesaurus generation studies relied on automatic indexing techniques [34]. Words were identified, stemmed, and combined to produce descriptors (automatic indexes of single or multiple keywords) to represent the content of a document. Despite its simplicity, automatic indexing may produce significant amounts of "noise," e.g., typos, meaningless acronyms, general terms, and random permutation of adjacent terms. Special attention needs to be paid to ensuring generation of specific and meaningful terms and a combination of techniques needs to be used. *Term fre-*

quency and *inverse document frequency* [34] are often required to reward terms that are specific.

- **Asymmetric association principle:**

  Human memory association is asymmetric by nature [2]. That is, the strength of the association from term A to term B is often different from the strength of association from term B to term A. This characteristic is also evident in information retrieval. However, this asymmetric association property of human memory and information retrieval had not been considered in most prevailing similarity functions. The limitation of the popular symmetric similarity functions, e.g., cosine, Dice, and Jaccard's, have been reported recently by Peat and Willett [30]. Their research showed that similar terms identified by symmetric co-occurrence function tended to occur very frequently in the database that is being searched and thus did little or nothing to improve the discriminatory power of the original query.

- **Relevance feedback principle:**

  Croft and Das [16] reported significant improvements in effectiveness of expanded queries when users are prompted for additional terms that can be used in the search. Automatic term replacement or switching is often misleading and impractical, considering the unique context and backgrounds that different searchers might have.

- **Vocabulary overlapping principle:**

  Numerous investigators in information retrieval have suggested the idea of "switching" languages, which could be consulted either automatically or manually, to aid searchers in performing multiple-database searches. Lancaster, in discussing compatibility and convertibility of vocabularies between databases, contended that because controlled

vocabularies tend to promote internal consistency within individual databases and information systems, they often reduce compatibility between systems [25]. Lancaster suggested that a "neutral switching language" can be used to convert from any one vocabulary into another. For scientific collaboration and information sharing across different domains, multiple domain-specific thesauri (existing or automatically generated) need to be created and coupled in order to assist in cross-domain term switching.

- **Spreading activation principle:**

  During the course of designing several large-scale, domain-specific thesauri, we found that the most frequent complaints from users who performed term switching manually (i.e., in a user-controlled browsing mode) was that the process was tedious and cognitively demanding and that users often got lost after exploring a number of concepts.

  Causes of such problems may be related to the second-order association described by Lesk [26]. Some terms may be related indirectly via their common associations with another term. Humans often perform such multiple-link associations (e.g., A is related to B, which in turn is related to C; or C is related to both A and B) during problem solving or long-term memory recall, a process frequently referred to as *spreading activation* [2]. Both Kim and Kim [24] and Chen et al. [11] proposed treating a thesaurus as a neural network or semantic network and applying spreading activation algorithms. Our recent experiment revealed that activation-based term suggestion was comparable to the manual thesaurus browsing process in document recall and precision, but that the manual browsing process was much more laborious and cognitively demanding [13].

Although the above concept space principles appear important and relevant based on our own experience and previous research, we recognize that they may serve only as guidelines for system development. More systematic and fine-grained testings of these principles would

be needed to provide a sound theoretical basis.

2.3 The Concept Space Techniques

Based on the seven principles described above, we developed selected algorithms for automatic thesaurus generation. The specific steps and algorithms adopted include: *document and object list collection*, *object filtering and automatic indexing*, *cluster analysis*, and *associative retrieval*.

We present below a brief overview of these techniques in the context of our experiment. For algorithmic details, readers are referred to [10] [11] [14] [13].

- **Document and object list collection:**

  In any automatic thesaurus building effort, the first task is to identify complete and recent collections of documents in specific subject domains that can serve as the sources of vocabularies. The proliferation of Internet services and the availability of online bibliographic databases have made document collection much easier.

  In [4], Bates proposed a design model for subject access in online catalogs. She stressed the importance of building domain-specific lexicons for online retrieval purposes. A domain-specific, controlled list of keywords can help identify legitimate search vocabularies and help searchers "dock" on to the retrieval system. For most domain-specific databases, there appear always to be some existing lists of subject descriptors (e.g., the subject indexes at the back of a textbook), researchers' names (e.g., author indexes or researchers' directories), and other domain-specific objects (e.g., genes, experimental methods, organizational names, etc.) which exist online or can be obtained through OCR scanning.

- **Object filtering and automatic indexing:**

For each online document, we first identified terms that matched with terms in our known vocabularies, a process referred to as *object filtering*. Because the texts remaining after object filtering may still contain many important concepts, an automatic indexing procedure, which included dictionary look-up, stop-wording, word stemming, and term-phrase formation, then followed [34].

- **Cluster analysis:**

  After terms were identified in each document, we first computed the term frequency and the document frequency for each term in a document. Term frequency, tf_{ij}, represents the number of occurrences of term j in document i. Document frequency, df_j, represents the number of documents in a collection of n documents in which term j occurs. A few changes were made to the standard *term frequency* and *inverse document frequency* measures.

  Usually terms identified from the title of a document are more descriptive than terms identified from the abstract of the document. In addition, terms identified by the object filters are usually more accurate than terms generated by automatic indexing. In our research, terms identified in titles were assigned heavier weights than terms in abstracts and terms identified by object filtering were assigned heavier weights than terms identified by automatic indexing.

  We then computed the combined weight of term j in document i, d_{ij}, based on the product of "term frequency" and "inverse document frequency" as follows:

$$d_{ij} = tf_{ij} \times \log(\frac{N}{df_j} \times w_j)$$

  where N represents the total number of documents in a collection and w_j represents the

number of words in descriptor j. Multiple-word terms were assigned heavier weights than single-word terms because multiple-word terms usually conveyed more precise semantic meaning than single-word terms.

We then performed term co-occurrence analysis based on the asymmetric "Cluster Function" developed by Chen and Lynch [10].

$$W_{jk} = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \times WeightingFactor(k)$$

$$W_{kj} = \frac{\sum_{i=1}^{n} d_{ikj}}{\sum_{i=1}^{n} d_{ik}} \times WeightingFactor(j)$$

W_{jk} indicates the similarity weights from term j to term k and W_{kj} indicates the similarity weights from term k to term j. d_{ij} and d_{ik} were calculated based on the equation in the previous step. d_{ijk} and d_{ikj} represent the combined weight of both descriptors j and k in document i. However, they were computed slightly differently due to their different starting terms. They are defined as follows:

$$d_{ijk} = tf_{ijk} \times \log\left(\frac{N}{df_{jk}} \times w_j\right)$$

$$d_{ikj} = tf_{ijk} \times \log\left(\frac{N}{df_{jk}} \times w_k\right)$$

where tf_{ijk} represents the number of occurrences of both term j and term k in document i (the smaller number of occurrences between the terms was chosen). df_{jk} represents the number of documents (in a collection of N documents) in which terms j and k occur together. w_j represents the number of words of descriptor j and w_k represents the

number of words of descriptor k (thus descriptors with multiple words receive higher weights).

In order to *penalize* general terms (terms which appeared in many places) in the co-occurrence analysis, we developed the following weighting scheme which is similar to the *inverse document frequency* function:

$$WeightingFactor(k) = \frac{\log \frac{N}{df_k}}{\log N}$$

$$WeightingFactor(j) = \frac{\log \frac{N}{df_j}}{\log N}$$

Terms with a higher df_k or df_j value (more general terms) had a smaller weighting factor value, which caused the co-occurrence probability to become smaller. In effect, general terms were *pushed* down in the co-occurrence table (terms in the co-occurrence table were presented in reverse probabilistic order, with more relevant terms appearing first).

- **Associative retrieval:**

  In addition to the user-controlled thesaurus browsing process, searchers can also invoke selected spreading activation algorithms for multiple-term, multiple-link term suggestions. We have developed two algorithms, based on the serial branch-and-bound algorithm and the parallel Hopfield net algorithm, respectively [13]. The Hopfield algorithm, in particular, has been shown to be ideal for concept-based information retrieval.

The Hopfield net [23] was introduced as a neural network that can be used as a content-addressable memory. Knowledge and information can be stored in single-layered, interconnected neurons (nodes) and weighted synapses (links) and can be retrieved based on the Hopfield network's *parallel relaxation* and *convergence* methods. The Hopfield net has been used successfully in such applications as image classification, character recognition, and robotics. Each term in the network-like thesaurus is treated as a neuron and the asymmetric weight between any two terms is taken as the unidirectional, weighted connection between neurons. Using user-supplied terms as input patterns, the Hopfield algorithm activates their neighbors (i.e., strongly associated terms), combines weights from all associated neighbors (by adding collective association strengths), and repeats this process until convergence. During the process, the algorithm causes a *damping effect*, where terms farther away from the initial terms receive gradually decreasing activation weights and activation eventually "dies out." This phenomenon is consistent with the human memory *spreading activation* process.

The Hopfield net algorithm relies on an activation and iteration process, where

$$\mu_j(t+1) = f_s[\sum_{i=0}^{n-1} W_{ij}\mu_i(t)],\ 0 \leq j \leq n-1$$

$\mu_j(t+1)$ is the activation value of neuron (term) j at iteration $t+1$, W_{ij} is the co-occurrence weight from neuron i to neuron j, and f_s is the continuous SIGMOID transformation function, which normalizes any given value to a value between 0 and 1. This formula shows the *parallel relaxation* property of the Hopfield net. (Readers are referred to [13] for algorithmic detail.)

We have implemented the above algorithms in C on various UNIX workstations (e.g., DEC Alpha 2100, HP 715/100). However, we believe that, in order to adopt the concept

space techniques in large-scale digital library collections, a parallel computing approach is needed.

3 A Parallel Computing Approach to Creating Engineering Concept Spaces

Over the past decade, we have seen parallel computing for information retrieval gradually move from an active research area to one of commercial potential [33]. Many new classes of algorithms and applications have emerged and created unique opportunities and challenges in parallel computing, especially in the context of Grand Challenge Applications and National Information Infrastructures [40].

3.1 Parallel Computing: From Information Retrieval to Knowledge Discovery

Parallel computing is defined as information processing that emphasizes concurrent manipulation of data belonging to one or more processes solving a single problem. Two classes of architecture have been used to distinguish between different parallel supercomputers: SIMD (single instruction stream, multiple data stream) and MIMD (multiple instruction stream, multiple data stream). In SIMD machines (e.g., MasPar MP-1), one control processor broadcasts a single instruction stream to all the other processors simultaneously for execution on different data streams. In MIMD machines (e.g., Thinking Machine's CM-5), each processor has its own independent program instruction stream. While this classification has been useful in distinguishing different supercomputing architectures, many other classification schemes have recently been proposed.

In [33], Rasmussen suggested three approaches to parallel computing in information retrieval (IR): development and testing of parallel IR algorithms, design of special-purpose parallel hardware for IR applications (e.g., database machines), and development of distributed systems for database access. The first approach, in particular, is of most relevance to our research.

A major focus of research in the 1980s has been the adaptation and refinement of existing popular IR algorithms to parallel processors. Pattern matching (string matching) algorithms, text signatures (superimposed coding) based retrieval algorithms, and inverted index file algorithms have attracted most of the attention [33]. Text signatures provide an efficient fixed-length document representation which is ideal for parallel processors [37]. However, Salton and Buckley [35] have shown that the limited memory units attached to the small processing units on Connection Machine cannot accommodate sophisticated term weights and performance degraded significantly. In [15], Couvreur et al. reported the results of modeling the performance of searching large text databases (10+ GBs of *Chemical Abstracts*) via various parallel hardware architectures and search algorithms. They found that a multiprocessor mainframe with parallel inverted index file algorithms and the TRW Fast Data Finder (FDF, special-purpose parallel IR hardware) with "on-the-fly" pattern matching capability out-performed loosely-coupled RISC processors with a text signature algorithm.

While parallelization of existing IR algorithms accounted for a majority of previous research efforts, a significant number of diverse and promising information processing and analysis algorithms have emerged and are believed to be ideal for parallel computation. Co-occurrence analysis and clustering algorithms have traditionally been the most computationally intensive algorithms in information science. As discussed previously, this class of algorithms often aims to compute "relationships" between term-pairs and/or document-

pairs and cluster terms/documents of similar nature. As Rasmussen commented [33], "An IR application that is particularly computationally intensive (usually $O(N^2)$ to $O(N^3)$, while offering a high degree of parallelism, is document clustering." He cautioned that the successful implementation of a parallel solution in IR requires an appropriate match of task, algorithm, and architecture.

Many new algorithms developed in the area of machine learning, in particular neural networks and genetic algorithms, are parallel in nature and have become prime candidates for parallel computation. Unlike the IR tasks performed by the previous pattern matching, signature files, and inverted index algorithms, most of these machine learning algorithms require extensive pre-processing and analysis of a significant number of textual documents. (Chen provides a complete and up-to-date review and discussion of machine learning techniques for IR in [7].)

Neural networks computing, in particular, seems to fit well with conventional retrieval models such as the vector space model [34] and the probabilistic model [28]. Oddy and Balakrishnan [29] described a parallel IR system in which a document collection is represented as a network mapped over a Connection Machine. The *PThomas* system is similar to the neural networks model conceptually. However, these researchers noted the practical problem of CM size limits (32K processors), which may render the approach to be infeasible for large-scale databases. They suggested network partition approach to solving this problem.

Yang and his coworkers [41] have developed adaptive retrieval methods based on genetic algorithms and the vector space model using relevance feedback. They reported the effect of adopting genetic algorithms in large databases, the impact of genetic operators, and GA's parallel searching capability. Frieder and Siegelmann [21] also reported a data placement strategy for parallel information retrieval systems using a genetic algorithms approach. Their results compared favorably with pseudo-optimal document allocations.

This emerging machine learning and information analysis paradigm for IR was also echoed in the recent "Report on Workshop on High Performance Computing and Communications for Grand Challenge Applications: Computer Vision, Speech and Natural Language Processing, and Artificial Intelligence" [40]. Automatic analysis of co-occurrence patterns in a text corpus, development of electronic librarians to locate information, and discovering new knowledge from existing sources are among the main areas of future research identified by the workshop participants. As the report stated: "High performance AI systems will undoubtedly require very large knowledge bases. Today, the construction of even small to medium knowledge bases is a very time-consuming process and often prevents the application of AI to real-world problems. To overcome this deficiency, automation of knowledge-base construction is needed. Knowledge may be acquired from the vast amount of information stored as texts. Patterns of concepts and their semantic properties may then be extracted from text via natural language parsing and learning techniques." Our previous and current work in the areas of automatic thesaurus generation, spreading activation, and machine learning based IR is a good example of adopting this new "knowledge discovery" paradigm for digital libraries.

3.2 Computer Engineering Concept Space Generation Using CM-5 and Power Challenge

In addition to discussing supercomputers based on the SIMD and MIMD architectures, some researchers also classify supercomputers in terms of their processor and memory requirements. According to Larry Smarr, director of NCSA, the first era of supercomputing belongs mainly to "shared memory vector processors" such as Cray X-MP, Cray Y-MP, Cray 2, and Convex C3. The second era of "distributed memory systems" include systems like CM-2, CM-5, IBM clusters, HP clusters, etc. More recently, "shared memory multiproces-

sors" (SMP) have emerged as the dominant force for the third and current era, e.g., SGI Power Challenge, Convex Exemplar, etc. The parallel computation implementation of the Illinois DLI semantic retrieval research coincided with the availability of supercomputers of the second and third eras. Our initial parallel implementation in Fall 1994 was based on a 512-node CM-5 and our recent testing was done mainly on the 16-processor SGI Power Challenge. Future parallel implementation of the Illinois DLI project will primarily involve a 48-processor SGI Power Challenge Array (SGIs), a 16-processor Cray CS6400 (SPARCs), and a 64-processor Convex Exemplar (HPs), all from NCSA.

- **Worm and fly concept spaces generation using CM-5 and SGI Power Challenge:**

  In our previous NSF-funded "Worm Community System" project, we adopted the concept space approach to cross-domain scientific information retrieval. By working closely with worm and fly biologists in the Molecular and Cellular Biology Department at the University of Arizona for about two years, we generated a worm thesaurus in Fall 1993 [14] and a fly thesaurus in Summer 1994. Both thesauri were independently tested by the biologists and are available for Internet WWW access at: http://bpaosf.bpa.arizona.edu:8000/cgi-bin/BioQuest.

  The resulting worm thesaurus consisted of 7,657 terms and 547,810 links and the fly thesaurus contained 15,626 terms and 750,314 links (after applying various thresholds). Most of these terms were author names or subject descriptors. The document collections were mainly Medline and Biosis abstracts acquired from online sources. Each collection contained about 7,000 abstracts, with 10 MBs of text. It took 50 and 70 minutes, respectively, to generate the two thesauri on a DEC Alpha 2100 workstation (200 MHz, 128-MB RAM). Automatic indexing, which is less computationally inten-

sive, took about 2.5 minutes; while the rest of the computation was for co-occurrence analysis. The resulting thesauri were about the same size as the initial document collections (i.e., 1 : 1 storage overhead).

In order to address the scalability issue for concept space generation for large collections, i.e., GB-scale collections, which are common in scientific and engineering domains (the topic domains of the Illinois DLI project), we proceeded to test all concept space algorithms, in particular those of automatic indexing and co-occurrence analysis on supercomputers at NCSA. Our initial platform was on CM-5.

The 7000+ abstracts of the fly collection were used to generate the same fly thesaurus using a 512-node CM-5. CM-5 was mainly based on the massively parallel architecture, with 512 SPARC Cypress processors on distributed memory. Its total memory size is 16 GBs (32 MBs/node on 512 nodes). In automatic indexing, we took the *data parallel* approach by breaking the data file into 100 documents (each 0.5 MB in size) and assigned each 100-document set to one node. Our existing C code was modified to C*, CM's data parallel programming language. In addition, each node needed to access a 2-MB *object filtering* file (with list of known biology terms). Due to the memory size limit of the CM-5 processing unit (node), the automatic indexing process on the same collection took about 19 minutes. However, the same *data parallel* approach worked very well for co-occurrence analysis. By sorting all unique terms in alphabetical order and partitioning them into 22 by 22 (484) chunks of ordered terms, we were able to assign each chunk (0.25 MB) to one CM-5 node. The small file size and processor scale-up greatly speeded up the processing time from 67 minutes (on the DEC Alpha) to 4 minutes, a 17-fold improvement (according to Thomborson [39], a more than 20-fold speedup on supercomputers is unlikely without expensive and time-consuming recoding). Although the testing was encouraging, CM-5 did not appear to be able to

Algorithm/Platform	DEC Alpha 2100	CM-5	SGI Power Challenge
Automatic Indexing	2.5 mins	19 mins	24 secs
Co-occurrence Analysis	67 mins	4 mins	21.5 mins
Total	69.5 mins	23 mins	22 mins

Table 1: A benchmark comparison summary on DEC Alpha, CM-5, and SGI Power Challenge

alleviate the automatic indexing bottleneck for large-scale test collections.

A further testing was conducted in Spring 1995 on the new NCSA Power Challenge (installed in Fall 1994), which is based on shared memory multiprocessor architecture. It contains 16 MIPS R8000 processors, with a total shared memory size of 4 GBs. Using the same *data parallel* approach, we fully utilized the 16 processors. The resulting processing time was 24 seconds for automatic indexing and 21.5 minutes for co-occurrence analysis. The shared memory architecture alleviated the automatic indexing bottleneck experienced in CM-5, but the co-occurrence analysis time was longer than that on CM-5. All programs were written in C. Table 1 summarizes the CPU time for the 10-MB fly collection on automatic indexing and co-occurrence analysis, respectively, on a DEC Alpha 2100, 512-node CM-5, and 16-processor SGI Power Challenge.

In summary, due to the (distributed) memory size limit of the CM-5 processing unit (32 MBs/node), the automatic indexing process, which involved a large object filtering file, became the bottleneck. However, the 512-node CM-5 was able to perform co-occurrence analysis (a similarity matrix computation by nature, an $O(N^2)$ process, where N is the number of terms) efficiently after we adopted a data parallel approach, where each node receives a small block (in size) of the matrix to compute. Due to the

large shared memory space (4 GBs), automatic indexing was no longer the bottleneck for SGI Power Challenge. However, co-occurrence analysis remained time-consuming, especially for the specific 16-node NCSA SGI Power Challenge we tested (due to a smaller number of processors and a slower clock rate).

The programming learning curve on SGI Power Challenge was significantly smoother than that on CM-5 (roughly 2 weeks vs. 2 months). After some careful consideration and discussions with researchers and staff at NCSA, SGI Power Challenge appeared most promising because of its ease of programming, large shared memory, and expandability. With the planned addition of the 48-processor SGI Power Challenge Array (and faster processors) and supercomputers of similar architecture (e.g., 64-processor Convex Exemplar, and 16-processor Cray CS6400), at NCSA, SMP-based parallel computers were our choice for further experiments.

Our most recent experiment involved a 24-node Convex Exemplar, also provided by NCSA. The NCSA Convex Exemplar employed in September 1995 was a 3-hypernode SPP-1200 system, with 24 HP PA-RISC 7200 chips (processors), 4 GBs of physical memory, and 88 of GBs disk space with peak performance 240 MFLOPS per processor and 1.9 GFLOPS per hypernode. Given more processors and a higher clock rate per processor (compared with the 16-node SGI Power Challenge), automatic indexing for the same fly collection took 0.39 minutes and co-occurrence analysis took 1.46 minutes, both better than CM-5 and the 16-node SGI Power Challenge.

For large-scale analysis of textual collections, we believe the shared memory multiprocessors (SMP) such as SGI Power Challenge (the NCSA SGI Power Challenge was also recently upgraded) and Convex Exemplar are extremely promising. In a recent issue of *Science* [31], two U.S. supercomputing center directors have commended SMP highly for its architectural fit with the emerging data mining (knowledge discovery)

and digital library applications.

- **Computer engineering concept space generation using SGI Power Challenge:**

  In the current Illinois DLI project, we obtained an INSPEC test collection of 400,000+ (computer science and electrical engineering abstracts from the 1992-1994 INSPEC database in Spring 1995. This 2-GB testbed was used recently to generate a computer engineering concept space using the 16-processor SGI Power Challenge. The automatic indexing process for this gigabyte collection (about 200-fold increase in size over the fly/worm collection) took 1.5 hours. The most computationally intensive co-occurrence analysis took 23 hours. It used about 25% of the available CPU cycles on the NCSA Power Challenge for a three-week period (in fact it was the largest single user of the NCSA supercomputing resources at that time). The computer engineering concept space contained about 270,000 terms and 4,000,000+ links. We estimated that running the same program on our DEC Alpha 2100 workstation for the same INSPEC collection would take about 20-30 days of CPU time. This long turn-around time was decided to be infeasible due to the iterative nature of our system development and testing effort and the need for testing other even larger collections, e.g., 5 million Compendex engineering abstracts.

  We also obtained an online version of the INSPEC thesaurus, which contains 7,000+ terms in a classification hierarchy (mostly narrower term, broader term, related term relationships). Our initial analysis showed that the computer engineering concept space appears to contain finer-grained and newer concepts (terms) than the INSPEC thesaurus. On the other hand, the INSPEC thesaurus provides a richer classification structure of the conceptual relationships via its symbolic links than the concept space.

The computer engineering concept space was indexed (using WAIS indexing) and recently placed on WWW as an Internet server at: http://ai.bpa.arizona.edu/cgi-bin/csquest. Figure 1 shows the search results (related terms) using this server, called *CSQuest*, with a search term, "distributed artificial intelligence." We recommend that this server be used as an interactive computer term suggester for searching any computer-related bibliographical databases (e.g., INSPEC database) or Internet services (e.g., the CS Technical Report projects).

4 System Evaluation: A Concept Association Experiment

In order to examine the performance of the computer engineering concept space in capturing meaningful conceptual associations between terms, we conducted a concept association experiment in Summer 1995 using the INSPEC thesaurus as the benchmark for comparison. The experimental design was similar to that of those adopted in memory association [2] [10] and information retrieval experiments [14]. We present the experimental design and results generated in this section.

4.1 Experimental Design

We performed a two-phase experiment involving 12 subjects affiliated with the University of Arizona Management Information Systems Department, including two faculty members, and ten graduate degree candidates who had successfully completed course work in Artificial Intelligence, Databases, or Telecommunications/Networking. Prior to Phase 1, experimenters solicited from each faculty subject a list of 16 candidate terms from his domain that could be used as test descriptors. For each domain, we selected eight terms found in both the

Figure 1: CSQuest-suggested terms for “distributed artificial intelligence”

engineering concept space and the INSPEC thesaurus. One term was discarded because subjects objected to it, leaving a total of 23 test descriptors.

In Phase 1 (*Recall Phase*), each subject (both students and faculty) was asked to generate through a free association process as many related terms as possible in response to each test descriptor presented. This phase of the experiment called upon subjects' memory recall.

In Phase 2 (*Recognition Phase*), experimenters created randomized lists of associated terms for subjects to evaluate with regard to their relevance to the test descriptor, including 40 associated terms suggested by the Computer Engineering Concept Space, and all terms suggested by the INSPEC thesaurus. The concept space can offer significantly more than 40 terms; we selected the highest weighted 40 terms for evaluation (about 2 screenfuls of terms). The 12 subjects were then asked to evaluate each suggested term according to the Likert-like scale: "Irrelevant," "Somewhat Relevant," "Very Relevant." Terms considered too general were to be ranked as "Irrelevant." This phase of the experiment called upon the subjects' ability to recognize relevant terms. Human beings are more likely to recognize than to recall. The complete experiment lasted between 1.25 hours and 2.5 hours for each subject.

4.2 Experimental Results: Concept Recall and Concept Precision

In contrast to the *document* recall and precision measures typically used in information science research, we adopted *concept recall* and *concept precision* for evaluation. Instead of examining the number of relevant documents retrieved, we counted the number of relevant terms (concepts) identified by the concept space and the INSPEC thesaurus. They were computed as follows:

$$ConceptRecall = \frac{\text{Number of Retrieved Relevant Concepts}}{\text{Number of Total Relevant Concepts}}$$

$$ConceptPrecision = \frac{\text{Number of Retrieved Relevant Concepts}}{\text{Number of Total Retrieved Concepts}}$$

The number of *Retrieved Relevant Concepts* represented the number of concepts judged "Very Relevant" or "Somewhat Relevant" for each thesaurus. *Total Relevant Concepts* represented the target set of concepts that can be obtained through user-thesaurus interaction, and included all concepts generated by the subjects in Phase 1, as well as those additional unique concepts judged relevant from the computer engineering concept space and the INSPEC thesaurus from Phase 2. Graduate student subjects generated between 0 and 49 terms, with a mean of 7.83 terms. Faculty subjects generated between 5 and 30 terms, with a mean of 16.47 terms. Based on this target set of concepts, we examined the relevant concepts generated by each thesaurus to determine the *concept recall*. *Total Retrieved Concepts*, representing the total number of terms suggested by either thesaurus, was used to calculate *concept precision* levels. For the concept space, this value was always 40. The number of retrieved terms offered by the INSPEC thesaurus ranged from 2 to 38, with a mean of 10.391 terms. Two-sample t-tests were performed for *concept recall* and *concept precision*. Separate comparisons were made for each group of subjects (graduate students and experts).

The ten graduate student subjects, responding to each of the 23 test descriptors, generated a total of 230 data sets. The results for *concept recall* and *concept precision* are shown in Figure 2. The concept recall results indicate a sample size of 218, which resulted from 12 data sets in which subjects did not respond to the test descriptor presented and thus were assigned a *Retrieved Relevant Concepts* value of zero for the denominator. Concept recall for the automatically generated concept space was 69%, significantly greater than the 17.7%

recall value resulting from the INSPEC thesaurus. This result can be attributed to the fact that the concept space offered the subject a greater number of potentially useful and relevant terms. Of the total set of relevant terms for each test descriptor, approximately 70% came from the automatic concept space. This points to a major advantage of the automatically generated concept space – that it can offer the searcher a richer and more meaningful space for concept association and term suggestion.

Concept precision for the concept space was less than that of the INSPEC thesaurus (59.5% vs. 68.2%), a difference that was statistically significant (at a 10% significance level). That the precision for INSPEC thesaurus is not 100% can be explained by the fact that although terms in a manually generated thesaurus are carefully selected to represent a limited number of highly relevant terms, subjects typically considered broader or parent terms to be irrelevant, which lessened the number of potentially relevant terms within the set suggested. It was not unexpected, then, that the INSPEC thesaurus fared better on precision than did the concept space. Automatic indexing, the technique used in automatic thesaurus generation, not only generates useful, but also noisy terms. Thresholds can be applied to limit this effect, but cannot eliminate it. Therefore the concept space would be expected to contain more potentially irrelevant terms.

Similar results were obtained from the faculty subjects. These subjects responded to those terms relevant to their area of expertise; one responded to Artificial Intelligence terms, the other to Database and Telecommunication/Networking terms. All data sets were completed by the faculty subjects, resulting in a sample size of 23. The experts' concept recall for the automatic thesaurus was somewhat lower than that for the student subjects, indicating that the faculty members' criteria for relevance was more stringent than that of the students. In addition, experts tended to have a much higher rate of matching thesaurus and concept space suggested terms. So, while they identified fewer terms as being relevant than did

```
                                  INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL            N      MEAN     STDEV  -+---------+---------+---------+-----
Rec Auto Stud   218    0.6908    0.1852                                    (*-)
Rec Insp Stud   218    0.1771    0.1381   (*)
                                         -+---------+---------+---------+-----
   POOLED STDEV =     0.1633           0.16      0.32      0.48      0.64


                                  INDIVIDUAL 95 PCT CI'S FOR MEAN
  LEVEL           N      MEAN     STDEV  -+---------+---------+---------+-----
Pre Auto Stud   230    0.5950    0.2822   (--------*--------)
Pre Insp Stud   230    0.6822    0.4153                    (--------*---------)
                                        -+---------+---------+---------+-----
   POOLED STDEV =     0.3551          0.550     0.600     0.650     0.700
```

Figure 2: ANOVA analysis for recall and precision with graduate student subjects

the student subjects, the presence of numerous matching terms from Phase 1 resulted in lower recall. Experts' concept precision is higher than that for the students for both the concept space and the INSPEC thesaurus, primarily because experts failed to respond to far fewer concepts than did the students. That the difference in precision performance was not statistically significant for the experts is probably attributable to the relatively small sample size.

In conclusion, the automatically generated computer engineering concept space performed much better than the INSPEC thesaurus with regard to concept recall, whereas the INSPEC thesaurus performed better than the concept space with regard to concept precision. The implications of these findings are that the concept space appears to be robust and useful, that the automatically-generated concept space and the manually-created INSPEC thesaurus complement one another, and that the greatest assistance to the searcher can be provided

```
                                  INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL          N      MEAN     STDEV  ----+---------+---------+---------+--
Rec Auto Fac   23    0.5923    0.1294                                (--*---)
Rec Insp Fac   23    0.1502    0.1064   (--*--)
                                       ----+---------+---------+---------+--
  POOLED STDEV =     0.1185              0.15      0.30      0.45      0.60


                                  INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL          N      MEAN     STDEV  --------+---------+---------+--------
Pre Auto Fac   23    0.6772    0.2004   (-------------*-------------)
Pre Insp Fac   23    0.7366    0.2665           (-------------*-------------)
                                       --------+---------+---------+--------
POOLED STDEV =       0.2358                 0.630     0.700     0.770
```

Figure 3: ANOVA analysis for recall and precision with faculty subjects

when both are available for query enhancement.

Although the concept association experimental result was encouraging, the usefulness of the concept space for improving user query results (i.e., document recall and precision) was not tested. In a recent experiment involving (nematode) worm and (Drosophila) fly concept spaces and a worm database, we have shown that concept spaces helped improve the document recall level (from 32% to 65%), although the document precision level did not improve significantly [12]. As the next step of our project, we plan to incorporate the engineering concept space in the Illinois engineering literature testbed and examine its usefulness in improving document recall and precision. Future experiments will also include several spreading activation techniques for consulting concept spaces algorithmically for document retrieval [13].

5 Conclusions and Discussions

This research presents preliminary results generated from the semantic retrieval research component of the NSF/ARPA/NASA-funded Illinois DLI project. Using a variation of the automatic thesaurus generation technique, which we refer to as the *concept space* approach, we aimed to create graphs of domain-specific concepts (terms) and their weighted co-occurrence relationships for all major engineering domains. Merging these concept spaces and providing traversal paths across different concept spaces could potentially help alleviate the *vocabulary (difference) problem* evident in large-scale information retrieval. We previously have successfully adopted such a technique for a smaller molecular biology domain (Worm Community System, with 10+ MBs of document collection) with encouraging results.

In order to address the scalability issue related to large-scale information retrieval and analysis for the current Illinois DLI project, we recently proceeded to experiment using the concept space approach on parallel supercomputers. Our test collection was 2+ GBs of computer science and electrical engineering abstracts and the concept space approach called for extensive textual and statistical analysis (a form of knowledge discovery) based on *automatic indexing* and *co-occurrence analysis* algorithms, both previously tested in the biology domain. Initial testing results using a 512-node CM-5 and a 16-processor SGI Power Challenge were promising. Power Challenge was later selected to automatically create a comprehensive computer engineering concept space of about 270,000 terms and 4,000,000+ links using 24.5 hours of CPU time. Our system evaluation involving 12 knowledgeable subjects revealed that the automatically-created computer engineering concept space generated significantly higher *concept recall* than the human-generated INSPEC thesaurus (concept space : INSPEC thesaurus = 69.08% : 17.71%). However, the INSPEC was more precise than the

automatic concept space (concept space : INSPEC thesaurus = 59.50% : 68.22%). Using the INSPEC thesaurus as the benchmark for comparison, we believe the computer engineering concept space has demonstrated its robustness and potential usefulness for suggesting relevant terms for search. However, we are convinced that multiple interfaces and multiple vocabulary search aids are necessary for effective concept-based search across multiple large-scale repositories and domains.

Our current work mainly involves: (1) creating concept spaces for other major engineering domains (roughly in the following order: chemical, materials, systems and industrial manufacturing, mechanical, aerospace, automatic, civil, agricultural and biosystems, geological and mining, marine, and nuclear and energy) using the 48-processor SGI Power Challenge Array, 16-processor Cray CS6400, and 64-processor Convex Exemplar (most accounts have been set up with NCSA already); and (2) developing robust graph matching and traversal algorithms for cross-domain, concept-based retrieval. Our ongoing experiment, which involves concept space generation for 5 million Compendex engineering abstracts, is currently based on a 64-processor Convex Exemplar provided by NCSA. Future work also will include generating individualized concept spaces for assisting in user-specific concept-based information retrieval. Results from our research will be incorporated into an operational SGML search interface for the Illinois DLI engineering testbed in 1996. We are also investigating methods by which this semantic retrieval capability might be extended and scaled up to large distributed repositories (the Net or the NII).

6 Acknowledgments

This project was supported mainly by the following grants: (1) NSF/ARPA/NASA Digital Library Initiative, IRI-9411318, 1994-1998 (B. Schatz, H. Chen, et. al, "Building the Inter-

space: Digital Library Infrastructure for a University Engineering Community"), (2) NSF CISE Research Initiation Award, IRI-9211418, 1992-1994 (H. Chen, "Building a Concept Space for an Electronic Community System"), (3) NSF CISE Special Initiative on Coordination Theory and Collaboration Technology, IRI-9015407, 1990-1993 (B. Schatz, "Building a National Collaboratory Testbed"), (4) AT&T Foundation Special Purpose Grants in Science and Engineering, 1994-1995 (H. Chen), and (5) National Center for Supercomputing Applications (NCSA), High-performance Computing Resources Grants, 1994-1996 (H. Chen).

We would also like to thank Dr. Anindya Datta, Pauline Cochrane, and Dr. Ann Bishop for their valuable suggestions, Jim Ashling of the Institution of Electrical Engineers (IEE, vendor of the INSPEC database) for providing the INSPEC thesaurus for our experiment, and Dr. Larry Smarr, Dr. Melanie Loots, and Mike Welge at NCSA for their kind assistance.

References

[1] T. Ahlswede and M. Evens. Generating a relational lexicon from a machine-readable dictionary. *International Journal of Lexicography*, 1(3):214–237, 1988.

[2] J. R. Anderson. *Cognitive Psychology and Its Implications, 2nd Ed.* W. H. Freeman and Company, New York, NY, 1985.

[3] J. R. Anderson. Indexing systems: extensions of the mind's organizing power. *Information and Behavior,*, 1, 1985.

[4] M. J. Bates. Subject access in online catalogs: a design model. *Journal of the American Society for Information Science*, 37(6):357–376, November 1986.

[5] N. J. Belkin, R. N. Oddy, and H. M. Brooks. Ask for information retrieval: Part I. background and theory. *Journal of Documentation*, 38(2):61–71, June 1982.

[6] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[7] H. Chen. Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science*, 46(3):194–216, April 1995.

[8] H. Chen and V. Dhar. Reducing indeterminism in consultation: a cognitive model of user/librarian interaction. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, pages 285–289, Seattle, WA, July 13-17, 1987.

[9] H. Chen and V. Dhar. Cognitive process as a basis for intelligent retrieval systems design. *Information Processing and Management*, 27(5):405–432, 1991.

[10] H. Chen and K. J. Lynch. Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man and Cybernetics*, 22(5):885–902, September/October 1992.

[11] H. Chen, K. J. Lynch, K. Basu, and D. T. Ng. Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE EXPERT, Special Series on Artificial Intelligence in Text-based Information Systems*, 8(2):25–34, April 1993.

[12] H. Chen, J. Martinez, D. T. Ng, and B. R. Schatz. A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the Worm Community System. *Journal of the American Society for Information Science*, forthcoming 1996.

[13] H. Chen and D. T. Ng. An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound vs. connectionist Hopfield net activation. *Journal of the American Society for Information Science*, 46(5):348–369, June 1995.

[14] H. Chen, B. R. Schatz, T. Yim, and D. Fye. Automatic thesaurus generation for an electronic community system. *Journal of the American Society for Information Science*, 46(3):175–193, April 1995.

[15] T. R. Couvreur, R. N. Benzel, S. F. Miller, and D. N. Zeitler. An analysis of performance and cost factors in searching large text databases using parallel search systems. *Journal of the American Society for Information Science*, 45(7):443–464, August 1994.

[16] W. B. Croft and R. Das. Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th Conference on Research and Development in Information Retrieval*, pages 349–365, Brussels, Belgium, 5-7 September 1990.

[17] C. J. Crouch and B. Yang. Experiments in automatic statistical thesaurus construction. In *Proceedings of the Fifteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 77–88, Copenhagen, Denmark, June 21-24 1992.

[18] L. B. Doyle. Indexing and abstracting by association. *American Documentation*, 13(4):378–390, October 1962.

[19] S. T. Dumais. Latent semantic indexing (LSI) and TREC-2. In *Text Retrieval Conference (TREC-2)*, pages 105–115, Bethesda, MD, November 4-6 1994.

[20] E. A. Fox, J. T. Nutter, T. Ahlswede, M. Evens, and J. Markowitz. Building a large thesaurus for information retrieval. In *2nd Conference on Applied Natural Language Processing, Association for Computational Linguistics*, Pages 101-108, Ballard, Bruce, Editor; Morristown, NJ: Bell Communications Research., 1988.

[21] O. Frieder and H. T. Siegelmann. On the allocation of documents in multiprocessor information retrieval systems. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 230–239, Chicago, IL, October 13-16, 1991.

[22] G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, November 1987.

[23] J. J. Hopfield. Neural network and physical systems with collective computational abilities. *Proceedings of the National Academy of Science, USA*, 79(4):2554–2558, 1982.

[24] Y. W. Kim and J. H. Kim. A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, 46:113–116, 1990.

[25] F. W. Lancaster. *Vocabulary Control for Information Retrieval*. Information Resources Press, Arlington, VA, 1986.

[26] M. E. Lesk. Word-word associations in document retrieval systems. *American Documentation*, 20(1):27–38, January 1969.

[27] D. A. Lindberg and B. L. Humphreys. The UMLS knowledge sources: Tools for building better user interface. In *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, pages 121–125, Los Alamitos, CA: Institute of Electrical and Electronics Engineers, November, 4-7 1990.

[28] M. E. Maron and J. L. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–243, July 1960.

[29] R. N. Oddy and B. Balakrishnan. PTHOMAS: an adaptive information retrieval system on the connection machine. *Information Processing and Management*, 27(4):317–335, 1991.

[30] H. J. Peat and P. Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383, June 1991.

[31] R. Pool. Off-the-shelf chips conquer the heights of computing. *Science*, 269:1359–1361, September 8 1995.

[32] E. Rasmussen. Clustering algorithms. In *Information Retrieval: Data Structures and Algorithms*, W. B. Frakes and R. Baeza-Yates, Editors, Prentice Hall, Englewood Cliffs, NJ, 1992.

[33] E. M. Rasmussen. Introduction: parallel processing and information retrieval. *Information Processing and Management*, 27(4):255–263, 1991.

[34] G. Salton. *Automatic Text Processing*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[35] G. Salton and C. Buckley. Parallel text search methods. *Communications of the ACM*, 31(2):202–215, February 1988.

[36] B. R. Schatz and J. B. Hardin. NSCA Mosaic and the World Wide Web: global hypermedia protocols for the internet. *Science*, 265:895–901, 12 August 1994.

[37] C. Stanfill and R. Thau. Information retrieval on the connection machine: 1 to 8192 gigabytes. *Information Processing and Management*, 27(4):285–310, 1991.

[38] H. E. Stiles. The association factor in information retrieval. *Journal of the Association of Computing Machinery*, 8(2):271–279, 1961.

[39] C. D. Thomborson. Does your workstation computation belong on a vector supercomputer? *Communications of the ACM*, 36(11):41–49, November 1993.

[40] B. Wah. Report on workshop on high performance computing and communications for grand challenge applications: computer vision, speech and natural language processing, and artificial intelligence. *IEEE Transactions on Knowledge and Data Engineering*, 5(1):138–154, February 1993.

[41] J. Yang and R. R. Korfhage. Effects of query term weights modification in document retrieval: a study based on a genetic algorithm. In *Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval*, pages 271–285, Las Vegas, NV, April 26-28, 1993.

List of Figures