Automatic Subject Indexing Using an Associative Neural Network

Yi-Ming Chung, William M. Pottenger, Bruce R. Schatz
CANIS - Community Systems Laboratory
University of Illinois at Urbana-Champaign, Champaign, IL 61820
{chung, billp, schatz}@canis.uiuc.edu
http://www.canis.uiuc.edu

ABSTRACT

The global growth in popularity of the World Wide Web has been enabled in part by the availability of browser based search tools which in turn have led to an increased demand for indexing techniques and technologies. As the amount of globally accessible information in community repositories grows, it is no longer cost-effective for such repositories to be indexed by professional indexers who have been trained to be consistent in subject assignment from controlled vocabulary lists. The era of amateur indexers is thus upon us, and the information infrastructure needs to provide support for such indexing if search of the Net is to produce useful results.

In this paper, we propose the *Concept Assigner*, an automatic subject indexing system based on a variant of the Hopfield network [13]. In the application discussed herein, a collection of documents is used to automatically create a subset of a thesaurus termed a *Concept Space* [4]. To automatically index an individual document, concepts extracted from the given document become the input pattern to a Concept Space represented as a Hopfield network. The Hopfield net parallel spreading activation process produces another set of concepts that are strongly related to the concepts of the input document. Such concepts are suitable for use in an interactive indexing environment.

A prototype of our automatic subject indexing system has been implemented as part of the *Interspace*, a semantic indexing and retrieval environment which supports statistically-based semantic indexing in a persistent object environment.

KEYWORDS: Automatic indexing, semantic indexing, semantic retrieval, automatic subject assignment, amateur indexing, Concept Space, information retrieval, Interspace, semantic locality

1 An Overview of Subject Indexing

Indexing has traditionally been one of the most important research topics in information science. Indexes facilitate retrieval of information in both traditional manual systems and newer computerized systems. Without proper indexing and indexes, search and retrieval are virtually impossible.

Meadow [22] observes that in library science, indexing records the values of various attributes expected to be used as a basis for searching. Simply put, the goal of subject indexing is to produce a set of attributes that represent the content or topics of a document[1]. Such attributes are, for example, “subject” or “author”. This set of attributes is later used in searches to retrieve the object. Since the goal of indexing is to provide for effective searching, the indexing process should be user-centered. Thus when Abstract & Indexing professionals select and assign keywords to a document, they also consider users’ information needs as well as aspects of the information of most interest to users.

Traditionally, a great deal of effort has been invested in subject indexing. Traditional human indexing has two main tasks [9]. The first is to recognize and select the essence, or “aboutness,” of a text. This is done by reading or scanning the document. The second task is to represent the essence of the text. In this process, the indexer assigns a set of index terms to represent the central topics of the document.

Ideally, an indexer reads the full text of a document before determining the “aboutness” of it. However, indexers typically work under severe time constraints. Cleverdon [5], for example, determined an optimum indexing time of four minutes, which makes reading the full text impossible in most cases. As a result, most indexers scan a document for its main topics.

While scanning, indexers normally engage both perceptual and conceptual faculties. Perceptual processes employ information based on the actual content of the document. Conceptual processes, on the other hand, use global knowledge not contained in the document itself, but rather in the knowl-

[1] A document here is defined broadly as an object that contains information. The representation can be in different media types such as text, image, video, or some combination of these.

edge that the author implies in the document or in the domain knowledge the indexer possesses. Models of conceptual and perceptual processes are presented in [8, 16, 21].

After determining the main concepts of a document, an indexer selects a set of conceptual terms to represent it. However, it is difficult to select a small set of "best" terms among all the possible terms that can represent a document. Many studies have found low agreement in indexer assignment of terms [1, 6, 31, 32]. In addition, Furnas, et. al. [10] showed that the probability that two amateur indexers will use the same term to classify an object is less than 20%. It is commonly known that a single term can have multiple meanings in different contexts and that a single concept can be represented using more than one term. In retrieval, therefore, term ambiguity and indexing inconsistency often lead to poor precision and poor recall.

Controlled vocabularies greatly improve inter-indexer consistency. A controlled vocabulary (often used synonymously with "thesaurus") usually defines conceptual terms and their relationships to each other in a particular domain of knowledge. To reduce ambiguity, each entry in such a thesaurus defines the scope and usage of a given term. Each entry also lists the given term's relationships with other terms, such as broader, narrower, and related, and USE and USED FOR relationships for synonym control. Some thesauri also have prior terms and SEE ALSO references. These semantic relationships help indexers find the best terms to represent a concept. In systems that use a controlled vocabulary, indexers must assign only terms from the controlled vocabulary to indexing records. This requires an indexer to be trained to select the "best" terms out of the vocabulary. However, the controlled vocabulary does not eliminate term selection problems, as it may be out of date in relation to the documents being indexed. Maintainers of controlled vocabularies spend a great deal of time keeping them up-to-date with the latest conceptual terms used in their respective domains of knowledge. While this effort can be quite expensive, the payoff comes in reducing later retrieval effort [9].

Although controlled vocabularies promote vocabulary consistency, most controlled vocabularies are general in nature due to the limitation on vocabulary size. To ensure consistency and specificity, some more modern information services, such as INSPEC and Compendex, adopt a hybrid system in which both controlled vocabulary indexes and natural language indexes are used. In these hybrid systems, besides the usual controlled vocabulary indexes, indexers select terms from the text to represent the specificity or new terminologies of the document, and place these terms in uncontrolled vocabulary indexes.

Manual indexing is expensive and time consuming, and there have been several attempts to develop techniques for automatic concept assignment. One of the attempts to assign subject indexes automatically was The Identification System (TIS), a content-based indexing system for news stories developed by Hayes and Weinstein in the early '90s [12]. A text categorization shell is built using if-then rules which take into account concepts identified in the text, where they appear in the text, and their respective "strengths" of occurrence. This system was applied to two of the four textual products of Reuters Ltd. for automatic categorization of news stories. The results of categorization were also used to index the documents.

A few years later, another project developed the NameFinder application to find occurrences of names in online text with specific features that simplify the identification of company names and the names of people [11]. NameFinder also attempted to solve the problem of name variation in that different names may be used to mean the same thing. NameFinder resolved this problem by automatically recognizing appropriate name variations while ignoring inappropriate ones. This was accomplished using built-in knowledge of the names as structured in a specific domain.

Although these two approaches perform fairly well in a single domain, it is not clear how well such approaches will scale across domains. Maintaining a large predefined set of categories can be problematic in such a case. NameFinder, for example, relies on built-in knowledge of the name structure in a specific domain. Domain-specific information retrieval applications such as these often suffer from significant reductions in precision when applied to larger scale information spaces which span multiple domains.

Additional approaches employing techniques to automatically classify documents are discussed in [20], [29], and [14]. Essentially, these techniques rely on the use of a predetermined knowledge base which captures the state of a subject area at a given point in time. However, such approaches lack scalability in that the creation of such knowledge bases is a highly labor-intensive process conducted by Abstracting & Indexing professionals.

In the future world of the Net, there will be large numbers of community repositories maintained by organizations which deal with specific domains of knowledge [27]. This trend can already be seen in the proliferation of web sites. It will not be cost-effective for such repositories to be indexed by professional indexers who have been trained to be consistent in subject assignment from controlled vocabulary lists. Nor will it be effective to derive multiple, domain-specific knowledge bases manually. Instead, the repositories will be maintained by subject matter experts who are only amateur indexers. These amateur indexers will need automatic support from tools which scale across domains in order to provide enough consistency in indexing for search facilities to be effective.

Our approach to automatic subject indexing thus provides scalable interactive indexing support for human subject matter experts who must classify documents in their domain. It approximates a controlled vocabulary list with an automatic "thesaurus" termed a Concept Space which is based on statistical term co-occurrence [4]. As opposed to labor-intensive human-generated knowledge bases, Concept Spaces are computed automatically based on collection content. Our system provides an analog of professional indexer consistency by computing a suggestion list of concept terms to assign to a particular document. Our hypothesis is that the context-based suggestion will limit the variability ordinarily seen in amateur indexers (improving the recall) while the subject experts will choose useful and correct classification terms (improving the precision).

In this paper we present an automatic concept assignment system, Concept Assigner, based on Concept Space and Hopfield network algorithms. The system is part of the on-going Interspace project, the flagship effort in the DARPA Information Management program being developed here at CANIS, the Community Systems Laboratory at the University of Illinois at Urbana-Champaign. The project as a whole is prototyping the information infrastructure necessary to support semantic indexing and retrieval in the future world of a billion community repositories [19, 27].

The Concept Assigner system design and algorithmic details are discussed in the following sections. In Section 2, we present an overview of the Interspace Prototype and review four major services of the Interspace. In Section 3, the algorithms, system design, and implementation of the Concept Assigner are discussed in detail. In Section 4 we present preliminary experimental results based on a study of the system, and depict test samples used to illustrate system performance. Finally, conclusions and future work are outlined in Section 5.

2 Overview of the Interspace Research Project

The Interspace research project is developing a prototype environment for semantic indexing of multimedia information in a testbed of real collections. The semantic indexing relies on statistical clustering for concepts and categories. Interactive navigation based on semantic indexing enables information retrieval at a deeper level than previously possible for large, diverse collections. We are in the process of developing algorithms for computing Concept Spaces, Category Maps, and Concept Assignment, and testing algorithm utility on engineering literature, map images, and medical literature. The Interspace Prototype will thus enable scalable, interactive semantic interoperability across subject domain, media type, and collection size.

The Interspace analysis environment seeks to unify disparate distributed information resources in one coherent model [26, 25]. It provides a rich set of operations in support of complex interoperable applications. Standard services include inter-object linking, remote execution, object persistence, and support for compound objects (usually referred to as compound documents).

The Interspace Prototype also serves as a testbed in which several large, real-world collections are available for experimental usage. To date, we have obtained and computed semantic indexes on the following collections: INSPEC, approximately 3 million abstracts across Computer Science, Electrical Engineering, and Physics; Compendex, approximately 2.6 million abstracts across all of engineering; MEDLINE, approximately 9.6 million medical abstracts; and Patterns, a community repository of a software engineering discussion list. In addition, in collaboration with the University of California at Santa Barbara DLI project, a large collection of geo-referenced aerial photographs form part of our testbed.

The Interspace Prototype is based upon a layered system design. Figure 1 shows the architecture of the Prototype. The service layer supports core functionality required by the kernel. The four major services are:

- Concept Space generation
- Category Map generation
- Multimedia Concept Extraction (MCE)
- Concept Assignment

These four services are managed by a Domain Manager. Domains are the connection points between the Service layer and the Kernel layer (Interspace Analysis). In brief, domains function to group items into distinct collections, and the Domain Manager invokes services to extract concepts (MCE), create Concept Spaces, generate subdomains (Category Maps), and assign concepts to items (Concept Assignment).

The Concept Assignment service interacts with both the MCE and the Concept Space service, and these two as well as the Category Map service are discussed briefly in the following sections. The Concept Assignment service is then discussed in detail in section 3.

2.1 Multimedia Concept Extraction (MCE) Service

The primary functionality provided by the MCE is the extraction of concepts from various types of multimedia information source units. The framework of the MCE is designed to support both text and image source units. Currently, a prototype of the MCE has been implemented with a foundation for processing text-based information sources. The MCE supports concept extraction from documents of various formats including, for example, SGML. It employs advanced natural language parsing techniques to identify noun phrases and then extracts said noun phrases and stores them as concept objects [2].

2.2 Concept Space Service

The purpose of the Concept Space service is to automatically generate domain-specific thesaurus subsets which represent the concepts and their associations in the underlying infor-

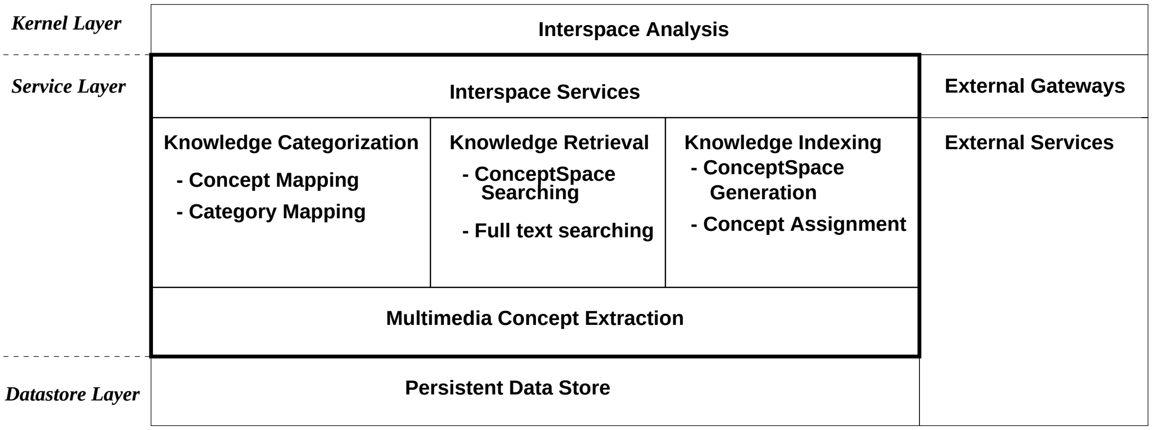

Figure 1: Interspace System Architecture

mation corpus. Concept Space generation is based on a statistical co-occurrence analysis which captures the similarity between each pair of concepts [3, 4]. The greater the similarity between concepts, the more relevant they are to one another. Concept Spaces are used in a retrieval environment to assist users in performing functions such as term suggestion [27, 28].

2.3 Category Map Service

The Category Map service classifies an information corpus into different conceptual categories using a variant of Kohonen's self organizing feature map (SOM) [18]. The SOM is a neural network which serves as vector quantizer to map high-dimensional feature vectors onto a two dimensional grid. A multi-layer Category Map is used to form a hierarchical set of categorizations for large corpora.

Several techniques are being incorporated in this service to assist the user in visualizing the hierarchical categorization. For example, the service will transform two-dimensional output maps into three-dimensional terrains which users can navigate via information spaceflight to investigate clusters of semantic locality [19].

3 Concept Assignment Service

In this section, we discuss details of the algorithm and system design and implementation of the Concept Assigner.

3.1 System Framework

Figure 2 depicts the interaction between the Concept Assigner and other Interspace Services. When a client requests that an item (document) be assigned key concepts, the Domain Manager initially invokes the MCE to extract concepts from the document.

Following this, the Domain Manger invokes the Concept Assigner and passes the concepts extracted by the MCE. The Concept Assigner then proceeds to assign concepts to the document using the Concept Space associated with the current domain as a repository of semantic patterns stored in a network.

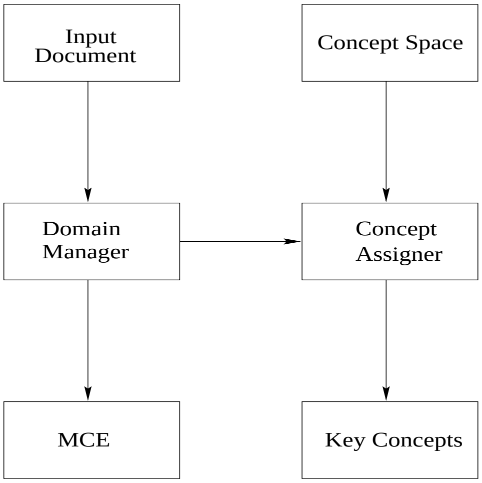

Figure 2: Concept Assigner Dataflow Diagram

In the following section, we outline the algorithmic steps in the process of automatic concept assignment.

3.2 Hopfield Net Algorithm Implementation

The Hopfield network [13, 30] was introduced as a neural net that can be viewed as content-addressable memory (CAM). Knowledge and information can be stored in single-layered interconnected neurons (or nodes) operating as fundamental memories which represent patterns stored in the network. This information can then be retrieved based on the network's parallel relaxation until the network reaches a stable state.

In this section, we review the Hopfield network implementation in the Interspace project. The Hopfield network has been adapted for the special needs of information retrieval. The network is an asymmetric, continuous network in which all the neurons are updated synchronously. The major steps of the algorithm are:

1. Assigning synaptic weights

   The Concept Space generated by similarity analysis serves as a trained network in the system. The concepts in the Concept Space represent nodes in the network and the similarities, computed based on co-occurrence analysis, represent synaptic weights between nodes (concepts). The synaptic weight from node i to node j is denoted w_{ij}. Note that the synaptic weights remain fixed after they have been assigned.

   To understand how a Concept Space is represented as a Hopfield network, consider the following example of a Concept Space represented as a semantic network:

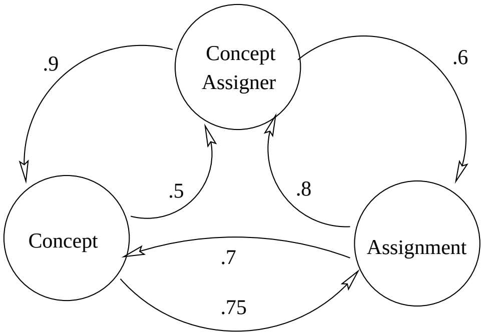

Figure 3: A Concept Space

   This network contains three nodes and six weighted, synaptic arcs connecting the nodes. The three concepts "Concept Assigner", "Concept" and "Assignment" all co-occur in the collection from which this example Concept Space was generated. The relationship between each pair of concepts (noun phrases) is expressed as the real-valued asymmetric similarity associated with each pair of concepts above. Effectively, this is the Hopfield network equivalent of a Concept Space.

2. Initialization

   An initial set of concepts (noun phrases) extracted from a document serves as the input pattern. Each node in the network matching one of the extracted concepts from the document is initialized to have a value of 1 (i.e., the node is activated). The rest of the nodes are assigned a weight value of 0 (i.e., they are deactivated). This is summarized in Equation 1 below:

$$u_i(t_0) = x_i, \qquad 0 \leq i \leq n-1 \tag{1}$$

   Here $u_i(t)$ is the output of node i at time t and x_i, with initial value 0 or 1, indicates the input pattern for node i.

3. Activation

$$u_i(t+1) = f_s\left[\sum_{j=0}^{n-1} w_{ji}u_j(t)\right], \qquad 0 \leq i \leq n-1 \tag{2}$$

   In the activation phase, n is total number of nodes in the network and f_s is the continuous sigmoid transformation function [7, 17] depicted below:

$$f_s(net_i) = \frac{1}{1 + exp\left[\frac{-(net_i - \theta_i)}{\theta_0}\right]} \tag{3}$$

   here $net_i = \sum_{j=0}^{n-1} w_{ji}u_j(t)$, θ_i serves as a threshold or bias, and θ_0 is used to modify the shape of the sigmoid function.

   This formula exemplifies the parallel relaxation property of a Hopfield net. At each time step, nodes in the Concept Space Hopfield net are activated in parallel, and activated values from neighboring nodes are combined for each individual node. The weight computation scheme, $net_i = \sum_{j=0}^{n-1} w_{ji}u_j(t)$, forms a unique characteristic of the Hopfield net algorithm in that each newly activated node computes its new weight (output state) based on the summation of the products of its neighboring nodes' weights and the similarity weight between its neighboring nodes and itself.

4. Convergence

   The above process is repeated until the network reaches a stable state, i.e., there is no significant change in the value of the output states between two time steps. To measure the stability of the network, the difference in activation level between two time steps is computed and compared to a predetermined convergence threshold, ϵ:

$$\sum_{i=0}^{n-1} |u_i(t+1) - u_i(t)| \leq \epsilon \tag{4}$$

   In this formula, ϵ is a threshold which indicates whether there is a significant global difference between two consecutive time steps across all nodes in the net. When the network converges, the final output is the set of concepts having the highest activation level. These concepts (terms or noun phrases, in the text domain) are considered most relevant to the concepts contained in the input document.

The algorithm outlined above can be intuitively understood as a process of determining in the Concept Space an area of semantic locality or a semantic cluster which encompasses a cluster of terms representing the input document. The semantic network pictured in Section 3.2 gives a trivial example of a cluster of terms exhibiting semantic locality in a Concept Space.

4 Experimental Results

We have conducted a preliminary study using a set of records drawn from the commercial Compendex collection. Compendex is a bibliographic database produced by Engineering Index which broadly covers all of engineering. The collection used for this study consists of 3 years (1993 to 1995) of Compendex records from the 723.1.1 classification (Programming Languages). Compendex Abstracting & Indexing professionals have assigned this classification to 7111 of the records in the database, and this is the size of the collection used in the experiments discussed herein. For our experiment, we first created a Concept Space for this collection and then used the Concept Assigner to assign concept descriptors automatically to individual documents.

To create a Concept Space for this domain, we used the following fields of each Compendex record as input to the similarity analysis: Title, Abstract, Author, Main Heading (MH), Controlled Vocabulary (CV), and Free Language (FL). MH, CV, and FL are indices selected manually by professional indexers at Compendex. MH is the main subject heading selected from among subject terms in the Compendex Ei Thesaurus, an engineering thesaurus developed by Compendex. The CV field contains terms drawn from a controlled vocabulary also represented in the Ei Thesaurus. A maximum of 12 controlled vocabulary terms can be assigned to a given record. FL are free language terms selected directly from the text and/or abstract of the article. Up to 10 FL terms can be included in a record.

The resulting Concept Space contained 43,813 concepts and 1,322,682 links (similarity relationships). The concepts and similarities in the Concept Space were used to initialize a Hopfield network of nodes (concepts/terms) and their weights (associations/similarities). The concept space was stored in an object-oriented database.

A subset of 290 records from the collection was used to perform Concept Assignment. The average size of the records was 1392 Bytes. In this experiment, we measured the process time needed for automatic concept assignment. The experiment was conducted on an Ultra 2 Model 2200 Sun with a 200 MHz UltraSPARC CPU and 256 MB of main memory. The measurement of processing time was broken down into two major processing tasks. The first task was to extract noun phrases from each abstract. The activities involved in this process included invoking a natural language parser to parse noun phrases as well as a search for matching noun phrases in the global database of existing concepts. This was computed by our Multimedia Concept Extraction service, and as shown in Table 1 took an average of about 4 seconds per record.

The second task was to traverse the Concept space to identify similar concepts using the Hopfield net algorithm described previously in Section 3.2. The average processing time is reported in Table 1, and as is shown, it took an average of just under 12 seconds of wall-clock time and 3 seconds of CPU time to automatically assign concepts to each record.

In total, processing for both steps took an average of less than 16 seconds wall-clock time. This is considered reasonable for an interactive application.

Table 1: Concept Assigner Service Processing Time

NP extraction		Net activation		Total	
CPU	W Clk	CPU	W Clk	CPU	W Clk
< 1	3.97	2.38	11.91	3.10	15.88

4.1 Detailed Example Runs

In this section, we present representative results for two abstracts for which concepts were assigned fully automatically by the Concept Assigner.

Each example run described below has been broken into eight fields which display both the input to and output of the Concept Assigner. The first seven fields are input, and the eighth, ”AUTOMATIC INDEXING”, is the output. The input fields are TITLE, JOURNAL, AUTHOR, ABSTRACT, Main Heading (MH), Controlled Vocabulary (CV), and Free Language (FL). As noted earlier, the MH, CV, and FL fields are assigned by Compendex Abstracting & Indexing professionals. All of these first seven fields (with the exception of the JOURNAL field) were used as input to the Concept Assigner. The eighth field, AUTOMATIC INDEXING, is the output from a fully automatic run of the Concept Assigner, and shows 30 index terms ranked in order which were suggested by the system.

4.1.1 Abstract Example 1 The two example abstracts are related to the process of parallelizing sequential programs for execution on multi-processor computers. The first example record is entitled *Idiom Recognition in the Polaris Parallelizing Compiler*. The first seven terms found by the Assigner are the names of researchers working on the Polaris project in the field of automatic parallelization. Such author name indices are relevant in searching for related work in the same subject area. The 10th term, “Detection of parallelism” and the 15th term “Automatic parallelization” are both highly relevant descriptors which represent the essence of the abstract. The 21st term, “Dependence analysis” and the 18th term, “Fortran (programming language)” are also relevant keywords for searching. (Automatically assigned relevant keywords are in boldface.)

With the exception of more general terms such as “Grand challenge” and “High-performance computing” (perhaps too general to be useful for discrimination in some searches), the system has successfully suggested several appropriate keywords. The key observation, however, is that the Concept Assigner has discovered an area of semantic locality in the Concept Space in that the two most relevant terms (“Automatic parallelization” and “Detection of parallelism”) *do not* *occur anywhere in the input record*. They have instead been drawn from other documents represented statistically in the Concept Space in a cluster of the terms used in the field of parallelizing compilers.

Example Abstract 1

TITLE: Idiom recognition in the polaris parallelizing compiler
JOURNAL: Proceedings of the International Conference on Supercomputing
AUTHOR: Pottenger, Bill and Eigenmann, Rudolf
ABSTRACT: The elimination of induction variables and the parallelization of reductions in FORTRAN programs have been shown to be integral to performance improvement on parallel computers. As part of the Polaris project, compiler passes that recognize these idioms have been implemented and evaluated. Developing these techniques to the point necessary to achieve significant speedups on real applications has prompted solutions to problems that have not been addressed in previous reports on idiom recognition techniques. These include analysis techniques capable of disproving zero-trip loops, symbolic handling facilities to compute closed forms of recurrences, and interfaces to other compilation passes such as the data-dependence test. In comparison, the recognition phase of solving induction variables, which has received most attention so far, has in fact turned out to be relatively straightforward. This paper provides an overview of techniques described in more detail in [23].
MH: Program compilers
CV: Parallel processing systems, FORTRAN (programming language) Algorithms, Data processing
FL: User interfaces, Polaris parallelizing compiler, Parallelization, Idioms recognition, Loops Induction variable, Induction, Data dependence

AUTOMATIC INDEXING:
1. Hoeflinger, jay; 2. Padua, david; 3. Eigenmann, rudolf; 4. Blume, william; 5. Petersen, paul; 6. Rauchwerger, lawrence; 7. Tu, peng; 8. Grand challenge; 9. High-performance computing; **10. Detection of parallelism**; 11. Automatic detection; 12. Cedar computer; 13. High performance computing; 14. Growing complexity; **15. Automatic parallelization**; 16. Parallel computer system; 17. Cedar fortran programming language; **18. Fortran (programming language)**; 19. Jaxon, greg; 20. Li, zhiyuan; **21. Dependence analysis**; 22. Fortran 77 programming language; 23. Conventional programming language; 24. Near future; 25. Brief description; 26. Fortran language; 27. Current implementation; 28. Multiprogramming; 29. Parallelism; 30. New technique

4.1.2 Abstract Example 2 In this example we will see a second case in which a cluster of concepts exhibiting semantic locality is discovered in the Concept Space. The abstract below deals with memory management in a parallelizing compiler. In this case, the Concept Assigner has identified fewer pertinent terms: specifically, only “Distributed memory parallel computer” can be considered highly relevant, and “Parallel fortran” and “Fortran (programming language)” relevant, although less so. The crux of the matter is that the cluster of terms exhibiting semantic locality centers around distinct but overlapping vocabularies.

Example Abstract 2

TITLE: Handling block-cyclic distributed arrays in Vienna Fortran 90
JOURNAL: Parallel Architectures and Compilation Techniques - Conference Proceedings
AUTHOR: Benkner, Siegfried
ABSTRACT: In this paper we describe the major techniques for parallelizing Fortran 90 array assignment statements for the execution on distributed memory parallel computers. We assume that the distribution of an array is specified by aligning it by means of a linear function with another array that is distributed in a block-cyclic manner across the processors of the parallel machine. We present techniques for the computation and representation of those elements of a distributed array that have to be allocated on a particular processor and for converting global indices into local indices. We show how the work distribution for assignments to regular array sections is derived automatically from the data distribution and present fast algorithms that determine the communication that is necessary between the processors of the parallel machine.
MH: Parallel processing systems
CV: FORTRAN (programming language), Distributed computer systems, Data storage equipment, Storage allocation (computer), Program processors, Computational methods, Algorithms
FL: Data communication systems, Block-cyclic distributed arrays, Distributed memory parallel computers, Distributed array, Data distribution, Parallel machine, Message passing

AUTOMATIC INDEXING:
1: Particle in cell; 2: Plasma simulation; 3: Decyk, viktor k.; 4: High performance scientific computation; 5: Machine-specific message-passing environment; 6: Norton, charles d.; 7: Memory parallel computer; 8: Cray t3d; 9: Plasma stability; **10: Distributed memory parallel computer**; 11: Object-oriented form; 12: Szymanski, boleslaw k.; 13: High-performance computing; **14: Parallel fortran**; 15: Parallel simulation; 16: Programming method; 17: Intel paragon; 18: Practical issue; 19: Parallel computer; 20: Parallel machine; 21: Parallel computation; 22: C^{++} program; 23: Language features; 24: Parallel processing systems; 25: Computer software portability; 26: Storage allocation (computer); 27: Natural sciences computing; 28: Software development; 29: Object-oriented programming; **30: Fortran (programming language)**

To understand this point, consider the following abstract also drawn from our Programming Languages collection and represented in the Concept Space:

Example Abstract 3

TITLE: Object-oriented parallel computation for plasma simulation
JOURNAL: Communications of the ACM
AUTHOR: Norton, Charles D., Szymanski, Boleslaw K. and Decyk, Viktor K.
ABSTRACT: Software development experiences with plasma Particle in Cell (PIC) simulation skeleton codes are discussed with the aim to evaluate object-oriented programming methods in high-performance computing. Beginning with the parallel Fortran 77 version, the application is converted into an object-oriented form using the Intel Paragon, IBM SP1/SP2, and Cray T3D distributed memory parallel computers. In addition, it is shown how Fortran 90 supports object-oriented programming by mirroring every language feature used in the sequential C^{++} program. In particular, this study focuses on the practical issues encountered in software development on parallel machines.
MH: Object oriented programming
CV: Parallel processing systems, Plasma simulation, Software engineering, FORTRAN (programming language), C (programming language), Natural sciences computing, Program compilers, Plasma stability, Computer simulation, Storage allocation (computer), Interfaces (computer)
FL: Computer software portability, Particle in cell, High performance scientific computation, Machine-specific message-passing environment, Parallel simulation

Both abstracts 2 and 3 are related to the following concepts: “Distributed memory parallel computer,” “Memory parallel computer,” “Parallelizing fortran” and “Communication.” It is reasonable that the concepts from these two documents should be closely related in the Concept Space. However, a second vocabulary is also involved in that abstract 3 discusses concepts related to the application of parallel programming techniques to a specific application area – namely “Plasma simulation”. The association between “Distributed memory parallel computer” and “Plasma simulation” thus results from the content of abstract 3. Clearly this association is not always desirable.

One potential solution to this problem is to deploy the Concept Assigner in an interactive environment in which the user can monitor and direct as necessary the activation process at any stage of the computation. Since it takes an average of only 12 seconds for the system to complete the computation, the process would be suitable for ‘live’ human interaction. An indexer, for example, could execute several runs of the system within the four minutes determined in [5]. If necessary, terms from a different vocabulary could be eliminated “on-the-fly” in this way. The indexer could also optionally assign higher weights to a particular term which he/she identified as an important keyword descriptor for the document. In this way, the Concept Assigner would function as an ‘intelligent assistant’ in helping the indexer identify key concepts describing a given document.

In the following section we present the results of a user evaluation study of the Concept Assigner.

4.2 User Evaluation Experiment

In order to evaluate the performance of the Concept Assigner, we conducted a user precision/recall evaluation experiment. Fourteen graduate students, half from Computer Science and half from Library and Information Science, participated in the evaluation. As noted in Section 4, the document database consisted of records drawn from the Compendex 723.1.1 classification, Programming Languages. A total of 76 documents were evaluated in the experiment discussed herein.

Participants in the study were given freedom to choose a subject domain of expertise within Programming Languages. A minimum of five documents dealing with the subject of their choice were evaluated. For each document, subjects were asked to first examine the title and abstract carefully and, based on their domain expertise, provide up to 20 index terms appropriate to the abstract. Subjects were free to choose such terms based on their own recall of relevant terms, the text of the title or abstract, etc.

Secondly, subjects were asked to evaluate a lexically ordered list of index terms consisting of terms assigned by Compendex professionals (the CV and FL fields in the abstract) combined with terms generated automatically by the Concept Assigner (AUTO). For each indexing term, subjects were required to judge the term as highly relevant, relevant, neutral, or not relevant to the “aboutness” of the document.

Term precision and term recall were used to measure the quality of each of the indexes: CV, FL, AUTO, and USER. Term precision is defined as the percentage of terms judged relevant out of the total number of terms in the given index. Term recall is defined as the percentage of terms judged relevant out of the total relevant terms. For the purposes of this study, we have defined total relevant terms as the sum of the relevant terms from each of the four indexes CV, FL, AUTO, and USER. Table 2 summarizes the results of this study.

Table 2: Concept Assigner Evaluation

	CV	FL	AUTO	USER
Precision	0.610	0.728	0.495	1.000
Recall	0.177	0.183	0.584	0.319

The fourth category in Table 2 represents user-selected keywords, and thus has the highest precision possible (100%). Of the remaining indexes, terms in the FL field were rated more precise (72%) than those in the CV field (61%). The automatically generated index resulted in a precision lower than the other three indexes (50%).

The three precision results for the CV, FL, and AUTO indexes are within the range expected for good retrieval performance [15]. In addition, we note that the precision of the CV and FL indexes is indicative of the competitive performance of natural language terms in indexing.

As expected, the recall of the Concept Assigner is higher than the three human-generated indexes (CV, FL, USER). A recall of 58% is more than thrice the recall of the professionally generated CV and FL indexes, and about twice that of the domain expert subjects. Of the four indexes, only the automatically generated keyword indexes fall within the range expected for good retrieval performance [15].

5 Conclusions and Future Work

From a practical perspective, the current technology may require human interaction to assure the quality of the resulting indexes - however, as discussed in the analysis of the results for Abstract 2 in Section 4, index terms can be automatically assigned very rapidly, thereby providing indexers with immediate feedback from an ‘intelligent assistant’ capable of deriving contextually determined semantics. Furthermore, as was noted in Section 1 “An Overview of Subject Indexing”, this capability will be in especially high demand in community repositories in which the curator is a subject matter expert but not a professional indexer, and the traditional publication process with its concomitant abstracting and indexing services rendered by professionals is simply unavailable.

From a technical perspective, ongoing research includes the determination of the precise mathematical conditions under which an asymmetric Hopfield net such as that employed herein will converge. Based on the Cholesky factorization, we can dynamically adjust the diagonal elements in the similarity weight matrix W to produce a non-negative definite matrix $\frac{1}{2}\sum_{j=1}^{n}|w_{ij}-w_{ji}| : i = 1,2,...,n \ \ and \ \ i \neq j$. This condition is sufficient to guarantee convergence by Theorem 1 in [33]. We are currently exploring the implementation and execution cost of a convergence test based on these criteria.

As noted in Section 4, the Concept Assigner performance is already such that it is reasonable to deploy it in an interactive environment. However, we are currently conducting indexing experiments on the entire National Library of Medicine MEDLINE collection, and advancements are needed to enhance performance when automatically indexing extremely large collections of this nature. Our previous research in the parallelization of the computation of Concept Spaces leads us to believe we can significantly improve the performance of concept assignment [24].

In the Net of many small collections, much of the emphasis of search will shift to searching across repositories. Concept Mapping of semantically related clusters of terms will then become important across the Concept Spaces of different collections. The ability to assign concepts to documents automatically for real collections naturally leads to technology for Concept Mapping. We are currently investigating spreading activation techniques for mapping concepts across spaces.

In conclusion, we believe the preliminary results discussed in this paper indicate that real potential exists for performing concept assignment automatically in various textual collections encompassing multiple domains of knowledge. The key to the scalability of these techniques lies in the domain-independent nature of the underlying statistical methods. The technology thus offers promise as a scalable technique for use across multiple domains of knowledge in an interactive environment as an ‘intelligent assistant’ for use by amateur indexers maintaining diverse community repositories on the Net.

6 Acknowledgments

Engineering Information, Inc. kindly provided the Compendex database as part of its partnership with the Digital Libraries Initiative at the University of Illinois at Urbana-Champaign. Financial support for this research was provided by the DARPA Information Management program under contract N66001-97-C-8535 and the NSF/DARPA/NASA Digital Libraries Initiative under cooperative agreement NSF IRI 94-11318COOP.

We would also like to thank Conrad Chang, Kevin Powell, Qin He, Nuala Bennett, Dan Pape, Margarita Ham, Ben Gross and Baba Buehler, and other members of the Interspace team both here and at the University of Arizona MIS Department for their time and effort without which this project would not have been possible.

REFERENCES

1. M. J. Bates. Subject access in online catalogs: A design model. *Journal of the American Society for Information Science*, 37:357–376, 1986.

2. N. Bennett, Q. He, C. Chang, and B. R. Schatz. Concept Extraction in the Interspace Prototype. *http://www.canis.uiuc.edu/interspace/technical/canis-report-0001.html*, 1997.

3. 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.

4. H. Chen, D. T. Ng, J. Martinez, 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*, 48(1):17–31, 1997.

5. C. Cleverdon. *Aslib Cranfield research project: report on the testing and analysis of an investigation into the comparative efficiency of indexing systems.* Cranfield, 1962.

6. W. S. Cooper. Is interindexer consistency a hobgoblin? *American Documentation*, 20:268–278, 1969.

7. J. Dalton and A. Deshmane. Artificial neural networks. *IEEE Potentials*, 10(2):33–36, April 1991.

8. J. F. Farrow. A cognitive process model of document indexing. *Journal of Documentation*, 47(2):149–166, June 1991.

9. R. Fugmann. *Subject Analysis and Indexing: Theoretical Foundation and Practical Advice*. INDEKS VERLAG, Germany, 1993.

10. 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.

11. P. Hayes and G. Koerner. Intelligent text technologies and their successful use by the information industry. In *Proceedings of National Online Meeting. 14th National Online Meeting*, pages 189–196, New York, N.Y., 1993.

12. P. Hayes and P. M. Weinstein. CONSTRU/TIS: A system for content-based indexing of a database of news stories. In *Innovative Applications of Artificial Intelligence 2*, pages 49–64, The AAAI Press/The MIT Press, Cambridge, Mass., 1991.

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

14. S. M. Humphrey. MedIndEx: the Medical Indexing Expert System. In *Expert systems in libraries*, pages 192–221, Rao Aluri and Donald E. Riggs, editors; Norwood, NL: Ablex, 1990.

15. K. S. Jones. *Information Retrieval Experiment*. Butterworths, 1981.

16. M. A. Just and P. A. Carpenter. *The psychology of reading and language comprehension*. Boston: Allyn and Bacon, 1987.

17. K. Knight. Connectionist ideas and algorithms. *Communications of the ACM*, 33(11):59–74, November 1990.

18. T. Kohonen. *Self-Organization and Associative Memory. Third Edition*. Springer-Verlag, Berlin Heidelberg, 1989.

19. CANIS Community Systems Lab. The Interspace Prototype. *http://www.canis.uiuc.edu/interspace.html*, 1998.

20. R. R. Larson. Experiments in Automatic Library of Congress Classificaiton. *Journal of the American Society for Information Science*, 43(2):130–148, March 1992.

21. M. E. Masson. Cognitive processes in skimming stories. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 8:400–417, 1982.

22. C. Meadow. *Text Information Retrieval Systems*. Academic Press Inc., San Diego, California, 1992.

23. W. M. Pottenger and R. Eigenmann. Parallelization in the Presence of Generalized Induction and Reduction Variables. Technical Report 1396, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., April 1995.

24. W. M. Pottenger and B. R. Schatz. Real-time Semantic Retrieval on Parallel Computers. Technical Report 1516, Univ. of Illinois at Urbana-Champaign, Center for Supercomputing Res. & Dev., October 1997.

25. B Schatz, W. Mischo, T. Cole, J. Hardin, A. Bishop, and H. Chen. Federating Diverse Collections of Scientific Literature. *IEEE Computer*, 29:28–36, May 1996.

26. B. R. Schatz. Building the Interspace: The Illinois Digital Library Project. *Communications of the ACM*, 38(4):62–63, April 1995.

27. B. R. Schatz. Information retrieval in digital libraries: Bringing search to the net. *Science*, 275(5298):327–334, Jan. 1997.

28. B. R. Schatz and H. Chen. Building Large-Scale Digital Libraries. *IEEE Computer*, 29(5):22–26, 1996.

29. K. Shafer. The Scorpion Project. *Accessible at: http://purl.oclc.org/scorpion*.

30. D. W. Tank and J. J. Hopfield. Collective computation in neuronlike circuits. *Scientific American*, 257(6):104–114, December 1987.

31. D. Tarr and H. Borko. Factors influencing interindexing consistency. In *Proceeding of the American Society for Information Science 37th Annual Meeting*, pages 50–55, 1974.

32. J. F. Tinker. Imprecision in meaning measured by inconsistency of indexing. *American Documentation*, 17:96–102, 1966.

33. Z. B. Xu, G. Q. Hu, and C. P. Kwong. Asymmetric Hopfield-type Networks: Theory and Applications. *Neural Networks*, 9:483–510, 1996.