COMPUTERS AND BIOMEDICAL RESEARCH 23, 410–425 (1990)

SAPHIRE—An Information Retrieval System Featuring Concept Matching, Automatic Indexing, Probabilistic Retrieval, and Hierarchical Relationships

WILLIAM R. HERSH AND ROBERT A. GREENES

Decision Systems Group, Harvard Medical School, Brigham and Women's Hospital, Department of Radiology, 75 Francis Street, Boston, Massachusetts 02115

Received August 30, 1989

SAPHIRE (Semantic and Probabilistic Heuristic Information Retrieval Environment) is an experimental computer program designed to test new techniques in automated information retrieval in the biomedical domain. A main feature of the program is a concept-finding algorithm that processes free text to find canonical concepts. The algorithm is designed to handle a wide variety of synonyms and convert them to canonical form. This allows natural language to be used for query input and also serves as the basis for a new approach to automatic indexing based on a combination of probabilistic and linguistic methods. © 1990 Academic Press, Inc.

INTRODUCTION

The information explosion in medicine is no longer recent news. Studies have shown that the information needs of physicians are not being met by conventional journals and textbooks (*1, 2*) and that a minority of physicians use the clinical literature in any substantial way (*3*). On a more philosophical level, the American Association of Medical Colleges has argued that the education of physicians places too much emphasis upon the rote memorization of medical information, while placing too little on problem-solving and the management of medical information (*4*).

There are many facets to the solution of the information overload problem, and some involve the use of digital computers, which have yet to reach their promise as clinical resource tools. A major problem is that computer solutions have been unable to get information to the physician needing it in a consistent and timely manner. This paper describes SAPHIRE (Semantic and Probabilistic Neuristic Information Retrieval Environment), an information retrieval capacity for Explorer-2, a system for knowledge management, and clinical decision support for clinical medicine (*5*).

SAPHIRE is an attempt to bring new functionality in automated information

0010-4809/90 $3.00

retrieval techniques to the biomedical domain. Some of the innovations are enhancements to existing methods, such as concept extraction and synonym substitution in both indexing and retrieval. Other features are more novel, most notably the use of automatic indexing and probabilistic retrieval methods with a semantic network-based controlled vocabulary. Most previous work with automatic indexing and probabilistic retrieval methods has been with systems using word-frequency-based methods and uncontrolled vocabularies.

Semantic networks were introduced as a knowledge representation method for artificial intelligence programs in the mid-1960s (*6*), but they have also proven to be useful for representing content in information systems (*7, 8*). A semantic network is a graphical depiction of the relationships among objects in a domain. The basic units are *nodes* and *links*. Nodes represent fundamental elements in the domain, such as terms or concepts, while links depict the relationships among them. What distinguishes a semantic network from other network representations is the presence of semantic meaning encoded in the links between nodes.

The most common type of link is an *is-a* link. This link indicates that one element is a member (or subclass) of a class of elements. A class typically has a collection of distinguishing properties. For example, the representation *human—(is-a)* → *mammal* states that *human* is a member (or subclass) of the class *mammal,* and has certain properties associated with the class. Other types of links can also be represented, and virtually any property of an object can be encoded as a link.

An important aspect of is-a links is *inheritance*. Properties of classes, which can be any defining characteristics, are inherited by members or subclasses of that class. This concept is important for representing domains that can be structured as taxonomies.

Each node in a semantic network has a semantic type, which is usually a superclass above the node in the hierarchy. The semantic type imparts restrictions upon a node, such as what relationships with other semantic type classes will be allowed. This constraining property is a crucial element in knowledge representation, since a property of knowledge is which relationships between concepts are not allowed.

The node-and-link structure of a semantic network is alternatively depicted as a collection of *frames*. The notion of frames was introduced by Marvin Minsky in 1975 (*9*). A frame is a more modular representation characterizing one node in a semantic network, which is described by a set of attributes or *slots*. Slots correspond to the set of link types connecting the node, in its semantic network representation, to other nodes. Associated with slots are *slot fillers,* the values of those attributes for a particular frame. Slot values correspond, in semantic network representation, to the names of other nodes to which the frame node is connected directly through links of the slot's type.

The controlled vocabulary that the semantic network of SAPHIRE is based upon is Meta-1, which comes from work done in the Unified Medical Language System (UMLS) project of the National Library of Medicine (NLM). Meta-1 is

the first version of an evolving metathesaurus that is derived from the union of several widely used medical indexing vocabularies (*10*). A major purpose of Meta-1 is to allow translation of concepts among vocabularies. It has a designated canonical representation for each concept (usually the existing MeSH term), and pointers to representations of that concept in other vocabularies. Meta-1 also features a network of semantic types, as well as semantic typing for each concept.

While not a fully instantiated semantic network, Meta-1 is a rich source of synonyms and hierarchical relationships which can be used to fill the synonym and is-a slots, respectively, of SAPHIRE's semanic network. As Meta-1 will not contain a thesaurus of nonmedical terms, other word synonyms will need to be added to provide robust thesaurus capabilities.

The first implementation of Meta-1 will be available in 1990 from the National Library of Medicine. In the meantime, the semantic network for SAPHIRE is based on the proposed structure of Meta-1, with concepts extracted from texts. Synonym and is-a relationships are derived from the Medical Subject Headings (MeSH) vocabulary, which is the NLM's current controlled vocabulary for indexing the medical literature (*11*).

This paper first outlines the main features of SAPHIRE, describing the rationale for their use. Next follows a description of the implementation of the program. The paper concludes by discussing some of the limitations of the current approaches taken and plans to overcome these in future work.

Features of SAPHIRE

Concept Matching

Humans ask for information in varying terms. A common complaint by end users of computer searching programs is that there is often rigidity in how the computer accepts terms for input. Few systems allow specification of concepts by their synonyms, the MEDLINE system of the National Library of Medicine being a notable exception (*12*). SAPHIRE is able to automatically handle synonym substitution at the word and concept level using the synonym slot from the semantic network. The user is able to enter free text queries and have the program convert synonyms to the canonical form of the concept used by the system vocabulary.

Word level synonyms substitute individual words that make up a concept. The relationship between the words is symmetric. For example, if *high* and *elevated* are designated as word synonyms, then both the phrases *high blood pressure* and *elevated blood pressure* will represent the same concept.

Concept level synonyms substitute one concept for another. Each synonym of a canonical form has a pointer to the canonical form. This is an asymmetric relationship, with all synonyms having one canonical form. For example, the canonical term *hypertension* has a concept synonym *high blood pressure*. Acronyms can also be considered concept synonyms, and thus *HTN* is a synonym to *hypertension* as well.

In addition to synonym substitution, SAPHIRE has two other features that allow varying ways for a concept to be expressed. The first is a stemming algorithm that removes common suffixes, including plurals, from terms. SAPHIRE uses the Porter stemming algorithm, which is easy to implement and provides comparable results with other stemming algorithms (*13*). The vocabulary is subject to stemming also, so that stemmed concepts from the query can be matched against the vocabulary, without the user having to take the process into account.

The second feature is a word completer, which will expand any fragment three characters or longer into all matching words. This frees the user from having to remember the exact spelling of words, allowing him or her to enter only the beginning letters of a word. The three-letter requirement is designed to prevent computational explosion when the matching words and synonyms are combined to find concepts.

All the above features are brought together under a concept-finding algorithm first developed by Shoval (*14*). Shoval's approach has been implemented in a different way from that originally described, in order to add the above features. The original algorithm was based on a semantic network with directed links. The lowest level nodes in the network represented individual words. Each of these nodes had *greater-than* links that pointed to nodes whose names were comprised of all the words (and synonyms to those words) that were part of the node name (such as *chest* and *x-ray* each having a greater-than link to *chest x-ray*). If these nodes were portions of nodes with even longer names (such as *chest x-ray* being a portion of *chest x-ray abnormality*), there were also greater-than links which pointed to the longer-named nodes.

In Shoval's algorithm, the system started by taking the words in the query and matching them to nodes in the semantic network. A process of spreading activation of the network was then initiated by expanding the nodes in the direction of their greater-than links. If a higher-level node could trace back from the greater-than links to all the word nodes that pointed to it, then it would become the active node and the initial nodes would become deactivated. The activation process would continue with the newly activated nodes, until the highest-level node could be found, which would represent the concept that subsumed the largest number of words in the original query.

SAPHIRE's concept-finding algorithm, which is described in detail with the rest of the implementation later, enhances Shoval's approach by providing a more generalized method of handling synonyms as well as adding a stemming algorithm and word completer. The entire process gives the user numerous varying ways to enter strings that will lead to the finding of canonical concepts. For example, the canonical term *hypertension* can be obtained from the strings *high blood pressure, elev blood pres,* and *HTN*.

The capabilities described so far can only take advantage of intravocabulary synonyms. Another major obstacle to information retrieval by computer is that although there currently exist many computerized medical databases, each unfortunately tends to utilize a unique vocabulary for indexing its content.

MeSH	SNOMed
Leukemia, Myelocytic	Myeloid Leukemias
Islands of Langerhans	Islets of Langerhans
Scleroderma, Systemic	Generalized Scleroderma

FIG. 1. Synonymous terms from Mesh and Snomed.

Although it is not used for bibliographic indexing and retrieval, an example of another medical vocabulary with many similar, essentially synonymous terms to MeSH is SNOMed (*15*). Figure 1 shows several examples of corresponding terms from MeSH and SNOMed. That this problem is a major obstacle to more widespread use of information systems by physicians has been recognized by the National Library of Medicine, which undertook the UMLS project several years ago to address this problem (*10*). A major impetus for the development of Meta-1 and its successors is to provide easier access to a wider variety of computerized databases.

The approach to concept matching described here is also designed to allow the intertranslation of concept synonyms among vocabularies. After the program has translated an input concept into Meta-1's canonical form, translation to equivalent terms in other vocabularies can be done. This will provide a mechanism for searching multiple information resources. Thus a user could search not only the knowledge bases of Explorer-2, for example, but also have the search terms converted to MeSH terms for searching the biomedical literature and to QMR (*16*) or DXPlain (*17*) terms for use by those expert system programs.

Automatic Indexing

Computer-based query use will not be more widespread until there is a critical mass of available information, so that the physician will expect to be able to find answers to the majority of his or her questions. One major bottleneck in the creation of large information resources has been in the indexing of content for rapid and accurate retrieval. Most existing computer-based information retrieval systems rely on human indexing of content, which is an expensive and laborious process (*18*). It has also been shown that there is significant variation in index term designation among human indexers (*19*).

Recognizing the problems of manual indexing, researchers in the field of Information Retrieval have attempted to create methods for automatic indexing of content. The major work in this area has been along two lines. The first has been the use of probabilistic techniques, based on word frequencies and document similarities, while the second has been through application of linguistic techniques in an attempt to recognize the meaning of concepts and relationships between them in a document.

Research utilizing probabilistic methods began in the 1950s. One early pioneer

in the field was Luhn, who observed that the best terms for indexing were words of intermediate frequency (*20*). He noted that extremely common words, such as *the* and *and*, occurred in every document and thus did not discriminate between documents. Likewise, words that occurred very rarely were also generally not good indexing terms.

Salton expanded on Luhn's work developing computational methods for choosing indexing terms. In particular, he developd a process of calculating the *term discrimination value,* which is a measure of how well a term discriminates one document from others (*21*). This was done by the creation of multidimensional document vectors, with each dimension consisting of a word in the document. The similarity of documents could then be calculated by taking the *cosine* of the angles between the vectors in multidimensional space. Queries into the document collection could likewise be put into vectors and compared with the vectors of documents, also using the cosine method to measure similarity.

The SMART system was implemented by Salton to perform experiments using these methods. The indexing process begins by extracting individual words. These words are initially compared against a stop list of common words, and if they are in the list are discarded. The words are then passed to a stemming algorithm that attempts to remove plural forms and common suffixes. Additional enhancements include the generation of automatic thesauri via clustering of words in common documents, as well as clustering of documents to improve efficiency as the system grows larger.

The results of the work of Salton and others have shown that automatic indexing could be as good, and sometimes better, than human indexing (*18*). Combined with the decreased cost due to lack of need for human indexers, this approach has appeal. Nonetheless, there are some shortcomings. The first of these is the problem of words that occur with low or moderate frequency which get designated as good indexing terms, but actually turn out not to impart any meaning as a descriptor of the text. An example of this would be a word like *teacher*, which, when mentioned in a medical text is most likely only peripherally related to the medical topic being described. Yet if a user query contained the word *teacher*, some documents retrieved might have relatively high matching scores because they contain the word, although it did not represent the major content of the document. This is because *teacher* might have a high indexing weight assigned in the indexing process, if it occurs in a moderate number of medical documents.

Another major limitation of word-frequency-based systems is that they do not take advantage of the meaning of words. In concepts words often serve as modifiers of other words. For example, the word *high* adds meaning when occurring in the phrase *high blood pressure*. But when a query features the word *high* in a different context, a word-frequency-based approach will still designate a match of the document.

These shortcomings have led to investigation into the use of linguistic methods in information retrieval work. This is the domain of natural language processing,

where systems are designed to handle morphological variation of words as well as semantic relationships among words that make up a concept. By recognizing the meaning of concepts, these systems have the potential to represent the content of text better than systems based on word frequencies. The MedSORT projects of Carbonell (*22*) and Evans (*23*) have been among the largest medical natural language processing projects to date. They have shown the ability to recognize a large number of variants of concepts in free text in several domains.

A major problem with present-day natural language systems is that they are still largely research tools that require a great deal of computing resources and run slowly even on expensive workstations. Salton argues that despite the theoretical benefit of linguistic technology, no one has shown that these systems perform better than word-frequency techniques (*21*).

SAPHIRE takes advantage of a linguistic methodology that falls short of full natural language processing, but is still able to recognize most concepts in text. This is the adaptation of Shoval's algorithm, which was described previously. SAPHIRE uses this approach to perform automatic indexing based on concepts, which are represented as frames in the semantic network, as opposed to words. Since the concept-finding algorithm recognizes a wide variety of synonyms, the indexing process captures concepts expressed in different ways but indexes them to the same canonical concept.

Vries used Shoval's approach to index aneuropathology database based on a vocabulary consisting of a semantic network devised for the neuropathology domain (*24*). Vries' system, however, did not utilize any measures of concept frequency. SAPHIRE, on the other hand, uses a combination of the probabilistic and linguistic approaches. It uses a frequency-based method to calculate a discrimination value for indexing terms, but uses an enhanced version of the Shoval approach so that only concepts in the vocabulary are indexed from the text and searched upon as query terms. The details of the implementation are described below.

Probabilistic Retrieval

Most automatic indexing in information retrieval systems is accompanied by probability-based retrieval. Most automated systems typically rank documents retrieved, based on weights assigned by the indexing process. This will hopefully bring the documents which most resemble the query to the top of the list, so they will be most accessible to the user. Probabilistic systems also attempt to allieviate the need for Boolean and other complicated search specifications. Borgman (*25*) has shown that many users have difficulty expressing their searches correctly using Boolean operators. Classic mistakes include the confusion of the meaning of AND with OR and performing an AND with an empty list. Salton *et al.* (*26*) have shown that most users tend to get better retrieval performance by simple specification of terms in a probabilistic system rather than by user-specified Boolean operations.

As in other probabilistic systems, the SAPHIRE user types a query into a

simple text box, after which SAPHIRE identifies the concepts and allows the user to choose which concepts are to be used in the actual search. SAPHIRE avoids Boolean searching entirely, instead using a document ranking system based on each term's discrimination value, which is described below.

Another advantage of probabilistic ranking of content matched arises from the observation that searching often tends to produce excessive amounts of retrieved information. SAPHIRE uses a ranking algorithm that gives a score to each item retrieved so that the user is returned a ranked list of items from which to choose. The approach is modeled after the ranking system of the IRX Project of the National Library of Medicine (*27*). However, whereas IRX uses a word-frequency approach and ranks documents based on occurrence of words in the query and those words' discrimination values, SAPHIRE assigns term weights based on the frequency of occurrence of the entire concept, or its synonyms. Since SAPHIRE calculates a discrimination value for each concept in the knowledge base, it can calculate a score for each document matching terms in the query.

Hierarchical Relationships

The converse problem to that of excessive retrieval is that searching can sometimes produce too little retrieved information. To help alleviate this problem, SAPHIRE utilizes hierarchical relations from its vocabulary. The user can add the children or parents of a term to the search list. SAPHIRE will take advantage of the Meta-1 system in this regard, whose hierarchies will be based largely on the hierarchies of the MeSH system.

If a user has done a search and finds too little content, or content that does not match the query well enough, he or she can select any or all of the search terms and add their parents or children to the search list. This process is presently under control of the user, although a future research agenda item is to investigate possible heuristics for performing expansions automatically.

The Current Implementation of SAPHIRE

SAPHIRE runs on the Macintosh II family of microcomputers. It is written mainly in Object Pascal (Apple Computer, Inc.), an object-oriented version of Pascal. Data management for the semantic network and indexing files is provided by C-Tree (Faircom Inc.), a package of B-Tree routines written in C and callable from Pascal in the Macintosh Programmer's Workshop (MPW) environment (Apple Computer, Inc.). At the present time, the textual content is stored in a Hypercard (Apple Computer, Inc.) stack, which is displayed by the program using HyperEngine (Symmetry Inc.), a tool for displaying Hypercard stacks from within programs written in conventional programming language.

SAPHIRE is a replacement for the existing keyword lookup section of Explorer-2 which currently relies on word fragment queries as input to Boolean search of author-selected keywords. One Explorer-2 knowledge base has been implemented in SAPHIRE, a hypertext implementation of the chapter *Acquired*

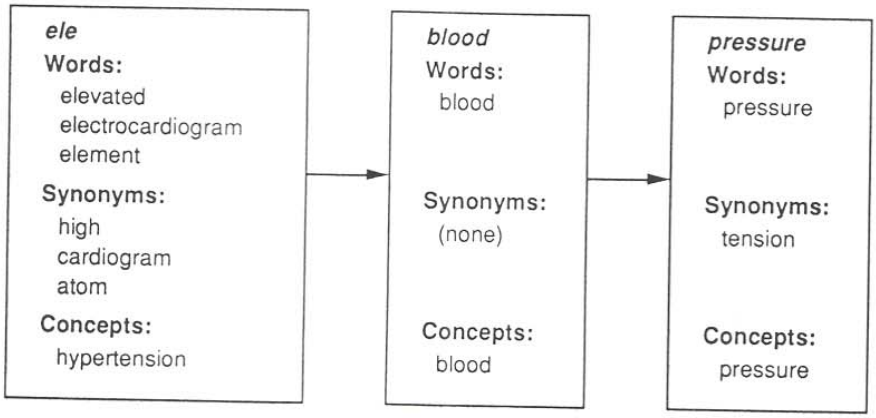

FIG. 2. Data structure for query *ele blood pressure*.

Immunodeficiency Syndrome from *Scientific American Medicine* (*28*). The vocabulary for indexing and retrieval was built manually, aided by some text processing routines described below. This was a very tedious process, and future vocabularies will be generated from other resources, particularly Meta-1 from the UMLS project of the NLM, which has been described above.

Concept-Finding Algorithm

The basic algorithm for extracting concepts is used both by the query processor and by the indexing routines. The process starts with creation of a list of all words or word fragments. Figure 2 diagrams the data structure for the algorithm. For each fragment, if it is in the vocabulary, or if it is a part of a multiword concept in the vocabulary, then the word is added to a word list for each item. If the actual word is not matched, but the fragment is three or more letters long, then all the words that start with the fragment are collected in a list. If this list is less than a certain length (currently seven words), then these words are added to the word list.

Behind the scenes, each word is also run through a stemming algorithm. SAPHIRE uses the Porter stemming algorithm (*13*). Stemming has been applied symmetrically across the vocabulary and indexing process. The process allows the program to handle varying suffixes and plurals.

The next step is to provide word level synonym substitution. Each item in the word list has a synonym list, and all one-word frames in the synonym slot for each word in the list are added to the list. A recursive backtracking algorithm is then performed to find the concepts that subsume the largest number of words. For each item in the word and synonym lists, the algorithm will attempt to combine it with all items in the following item's word and synonym lists, looking for concepts that contain or begin with the combination. If a word from the item and the item following it can be combined into a concept or into the words that begin another concept, then the process is repeated with the next item. This process continues until the words or synonyms can no longer be added to find concepts in the vocabulary.

As an example, consider the query *ele blood pressure*. The initial fragments are *ele, blood,* and *pressure*. Suppose the word list for *ele* contains all words that begin with *ele*, such as *elevated, electrocardiogram,* and *element*. Suppose also that the word list for *blood* and *pressure* just contains those words respectively. Word level synonyms added to the *ele* fragment will include the respective synonyms *high, cardiogram,* and *atom*. All of the words and synonyms of the *ele* fragment will be combined with the words and synonyms in the *blood* fragment (of which there will only be *blood*), and the only fragment found that is a concept or makes up the start of a concept is *high blood*. Likewise, combining *high blood* with the next fragment, *pressure,* leads to finding of the actual concept in the vocabulary *high blood pressure*. Since no longer concepts are found, *high blood pressure* is added to a match list.

The algorithm then requires repeated passes through the list using the same process, in order to find concepts that are made up of concepts that are synonyms of multiword concepts. An example of this is the concept *malignant hypertension,* whose latter term *hypertension* is a synonym of *high blood pressure*. If the user enters *malignant high blood pressure,* then the first pass will have passed by *malignant* when it was unable to find any concepts beginning with *malignant high*. In the second pass, *malignant* and *hypertension* will be adjacent concepts, and thus combined to form the concept *malignant hypertension*. This process continues until no more higher-level concepts are found.

Retrieval

A query is initiated by choosing the appropriate menu item, which brings forward the Query window, as shown in Fig. 3. The window consists of three panes. The upper pane is a text entry box for free text entry of the query. The middle pane shows concepts matched and allows the user to designate which concepts should be included in the actual search. The bottom pane shows the listing of knowledge frames which have matched the search terms.

The query process is begun by entering free text in the text entry box. Text is entered as whole words or fragments. Clicking the **Find** button causes processing to extract concepts from the text that subsume the largest number of words, allowing for synonym substitution, using the process described earlier.

After this process is completed, all concepts found are placed into a list in the middle pane. The user can clear out the text box for another query by clicking the **Clear** button in the upper pane.

The middle pane has two scrolling lists. The left list is for terms that have been matched from the free text entered. The user can highlight one or more of the terms in this list for removal, addition of parents, or addition of children. The highlighted terms can also be moved to the right list of the middle pane, which contains terms that the user actually wants to include in the search. These terms can likewise be highlighted for removal, addition of parents, or addition of children. The right list also contains a number in brackets, which indicates the number of documents in the collection which match the particular term.

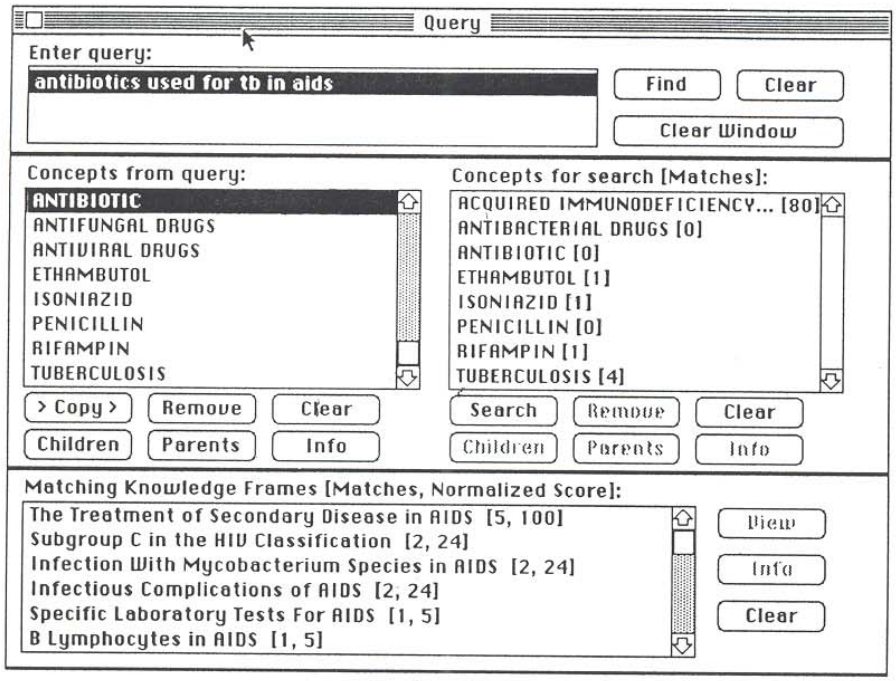

FIG. 3. Query window.

Clicking on the **Search** button will cause a search to occur based on all the terms present in the list. In the search process, every document which has at least one matching concept is put on the list of matching documents. The score for an individual document is calculated by summing the term weights of all the terms that match. The documents are then sorted by their scores, with the results displayed in the bottom pane. The user double clicks on the item that he or she wishes to view, with control transferred to the HyperEngine window containing the content.

The entire window operates in a modeless fashion. If the user moves to another window of the program, the query window does not disappear or change. When the user clears one portion of the window, the other parts remain the same. Thus if a user wishes to add new terms to the existing search terms, he or she does not have to start completely over. Both the **Find** and **Copy** buttons preform additive and not replacement actions.

In the sample search shown in Fig. 3, the user has begun by entering the free text *antibiotics used for tb in aids*. When the **Find** button is clicked, the concept-finding algorithm begins. The algorithm first stems *antibiotics* to find the canonical concept *antibiotic*. This is followed by continuing the processing of fragments until canonical concepts *Tuberculosis* and *Acquired Immunodeficiency Syndrome* are found by synonym translation. These concepts are added to the left scrolling list of the middle pane.

Once the concepts have been added to the left list, they can be selected and copied to the right list, which is the actual search list. In this case, the user has

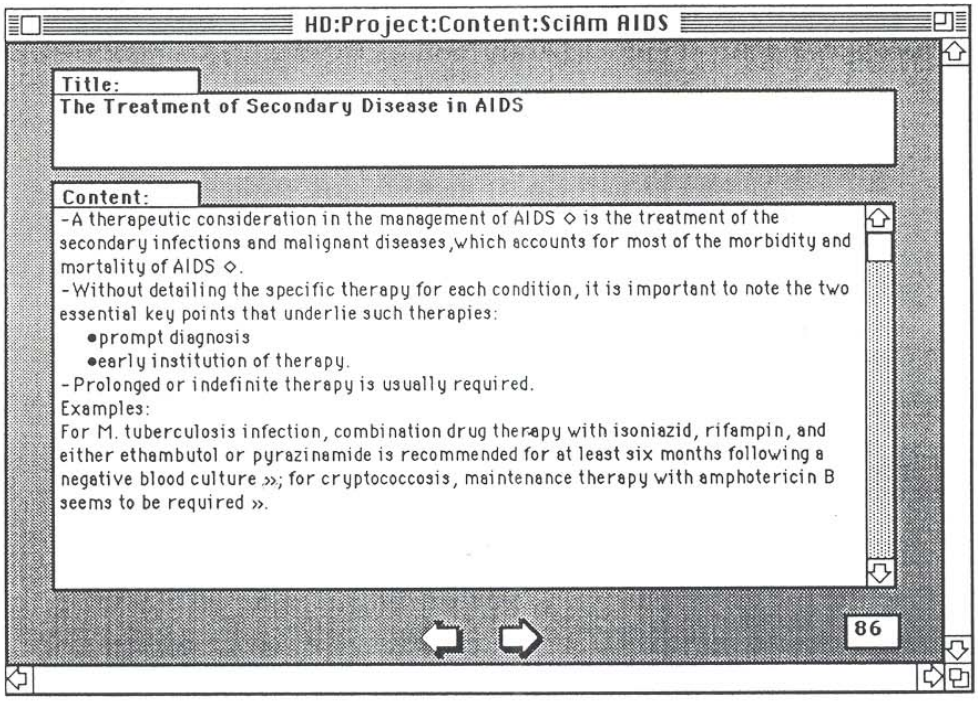

FIG. 4. Document retrieved by query in Fig. 3.

noted that the concept *antibiotic* has no matching documents. Thus in the left pane, *antibiotics* has been highlighted by the user and its children chosen. This causes the concepts *antibacterial drugs, antifungal drugs,* and *antiviral drugs* to be added to the list. But since no matching documents for these concepts exist either, further descent of the tree is necessary. The user has chosen *antibacterial drugs* and selected from its children the drugs *ethambutol, isoniazid,* and *rifampin.*

After the user is satisfied with the list of concepts for searching in the right middle pane, clicking the **Search** button causes the actual search to be performed. The matching hypertext documents appear in the scrolling list of the bottom pane. As can be seen, the top-ranked document has a score far above the rest. As can be seen by examining the matching document in Fig. 4, there are instances of *AIDS, tuberculosis, isonaizid, rifampin,* and *ethambutol.*

Indexing

The indexing process operates in a fashion similar to the searching process. Documents are broken up into sentences from which concepts are extracted nearly identically to the query process described above. As in the query process, synonyms are substituted and the concepts subsuming the largest number of words are matched. There are two differences, however, in the indexing process. First, all matching concepts are chosen as indexing terms, not just the

concepts that subsume the largest number of words. For example, the occurrence in the text of *carcinoma of the colon* will lead to the following concepts chosen for indexing: *carcinoma of the colon, carcinoma,* and *colon.* Second, the indexing process does not allow word fragments, since writers do not use them in their texts.

Once a concept has been found in the text, it is given a concept weight. For these weights, SAPHIRE uses the discrimination value, which is a measure of utility for a concept as an indexing term. There have been numerous proposed methods to calculate discrimination values; SAPHIRE uses the inverse document frequency:

$$\mathrm{IDF}_i = \log\left(\frac{N}{N_i}\right) + 1, \qquad [1]$$

where

IDF_i = inverse document frequency of term i
N = number of documents
N_i = number of documents where term i occurs.

Concepts which have a high discrimination value, and thus are deemed to be the best discriminators between documents as indexing concepts, are those concepts which occur in the fewest documents. For example, *multifocal leukencephalopathy* occurs in only two documents, and is thus a good concept for discriminating those two documents from the others in the collection. On the other hand, the concept *AIDS* occurs in four-fifths of the documents in the chapter, and is thus a poor discriminator and has a low discrimination value.

Vocabulary

The vocabulary for the AIDS chapter knowledge base used by SAPHIRE is a small semantic network based on concepts known to be present in the chapter. It was created by writing two small programs that processed the text. The first program created a file of all words from the chapter, while the second program created a file of all consecutive pairs, triples, quadruples, and quintuples of words in the chapter. The latter list was combed to create a file of all word combinations that were meaninfgul concepts. The files of single and multiword concepts were then combined, with synonyms and hierarchies designated for a small portion of the words. Figure 5 demonstrates a frame from the semantic network.

This approach to vocabulary building is obviously infeasible for all medical content, although the programs are useful for helping to identify medical content that occurs in text. The more long-term solution will be to use the Meta-1 vocabulary and its successors when they become available, which will provide a comprehensive list of concepts, their synonyms, and hierarchical relationships.

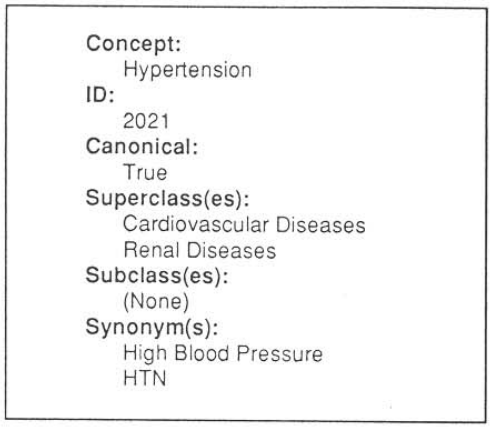

FIG. 5. A frame from the SAPHIRE semantic network.

CURRENT LIMITATIONS AND FUTURE PLANS

Initial experience with SAPHIRE has shown it to retrieve information relevant to queries entered by users. Because of the program's current small vocabulary and limited content material, large-scale evaluation is not able to be undertaken. Once Meta-1 is available and larger tracts of content have been procured for indexing, we intend to install the system in a clinical environment in Brigham and Women's Hospital (*29*) where its usage can be evaluated with both a log file of users actions as well as user feedback.

Not only is SAPHIRE presently limited in content, but it also has been tested in one type of information model, which is hypertext. For a comprehensive medical information system, one would want to devise an approach that can handle all types of information. Frisse (*30*) has demonstrated, for example, that information retrieval methods for hypertext differ from that of journal articles. SAPHIRE will be enhanced to allow translation of terms to other vocabularies for searching of other resources, such as MEDLINE and other currently available databases. Modifications will need to be made to the searching process when these other databases are used, since they have been indexed by manual methods and are designed for a Boolean searching paradigm.

An additional shortcoming is that the Porter stemming algorithm has also shown itself to be limited in a medical domain. There are certain suffixes that convey meaning in the medical domain, and whose removal leads to loss of specificity. For example, *hepatitis* and *hepatic* are both stemmed to *hepat*, despite the fact that the two words have different meaning. Furthermore, certain compound words, such as *bronchoscopy*, will need to be recognized and decomposed. Thus the algorithm will need to be modified to perform better in a medical information environment.

Also an area for improvement is in term weighting. At the present time, the weight for an indexing term is solely based on the inverse document frequency. Fuhr (*31*) has devised a word-frequency-based system that utilizes many other measures that contribute to a term's weight, such as whether it occurs in the

title and/or abstract, and how frequently the term occurs in the individual document. Just as several word-frequency-based system measures have been adapted for SAPHIRE's controlled vocabulary-based system, a future goal is to devise a more comprehensive probabilistic model for indexing using controlled vocabularies.

SAPHIRE is also limited in the amount of semantic meaning it can handle in its free text queries. Users may enter complex queries with many concepts and the relationships among them, yet only individual concepts are extracted from queries. The only relationships from the semantic network that the system now utilizes are synonym and is-a relationships. It is planned to investigate ways to handle other semantic relationships between concepts without affecting the ease of use of the program or its ability to do automatic indexing.

Conclusion

SAPHIRE is a prototypical new approach to information retrieval in the medical domain. It attempts to combine existing work in probabilistic and linguistic methods in the field of Information Retrieval, as well as adapt more recent work done as part of the Unified Medical Language System project of the National Library of Medicine. Future enhancements to SAPHIRE will aim to ultimately create an indexing and query system designed to handle many of the information needs of physicians.

Acknowledgments

This work was supported in part by Grant LM 07037 and Contract LM 63523 from the National Library of Medicine. The authors also gratefully acknowledge the advice and assistance of David B. Tarabar and Edward Pattison-Gordon of the Decision Systems Group.

References

1. Covell, D. G., Uman, G. C., and Manning, P. R. Information needs in office practice: Are they being met? *Ann. Intern. Med.* **103,** 596 (1985).
2. Williamson, J. W., German, P. S., Weiss, R., Skinner, E. A., and Bowes, F. Health science information management and continuing education of physicians. *Ann. Intern. Med.* **110,** 151 (1989).
3. Huth, E. J. The underused medical literature. *Ann. Intern. Med.* **110,** 99 (1989).
4. Muller, S. (Chairman). Physicians for the twenty-first century: The GPEP report. (Report of the Project Panel on the General and Professional Education of the Physician and College Preparation for Medicine). *J. Med. Educ.* **59,** (1984).
5. Greenes, R. A., Tarabar, D. B., Cope, L., Slosser, E., Hersh, W. R., Pattison-Gordon, E., Abendroth, T., Rathe, R., and Snydr-Michal, J. Explorer-2: An object-oriented framework for knowledge management. "Proceedings of the Medinfo 89." North Holland, Amsterdam, 29–33.
6. Brachman, R. J. On the epistemological status of semantic networks. *In* "Readings in Knowledge Representation" (R. J. Brachman and H. J. Levesque, Eds.), pp. 191–216.
7. Fox, E. A. Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. *Inf. Process. Manage.* **23,** 341 (1987).
8. Cohen, P. R., and Kjeldsen R. Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage.* **25,** 255 (1987).

9. MINSKY, M. A framework for representing knowledge. *In* "The Psychology of Computer Vision" (P. H. Winston, Ed.), pp. 211–277. McGraw–Hill, New York, 1975.
10. HUMPHREYS, B. L., AND LINDBERG, D. A. B. Building the unified medical language system. *In* "Proceedings of the 13th Annual SCAMC." IEEE, New York, 1989, 475–480.
11. "Medical Subject Headings—Tree Structures, 1989." National Library of Medicine, Bethesda, MD, 1988.
12. "Medline: A Basic Guide to Searching." National Library of Medicine, Bethesda, MD, 1985.
13. PORTER, M. F. An algorithm for suffix stripping. *Program* **14,** 130 (1980).
15. "Systematic Nomenclature of Medicine." 2nd ed. College of American Pathologists, Skokie, IL, 1982.
16. MILLER, R. A., MCNEIL, M. A.,CHALLINOR, S. M., MASARIE, F. E., AND MYERS, J. D. The Internist-1/Quick Medical Reference Project—Status report. *West. J. Med.* **145,** 816 (1986).
17. BARNETT, G. O., CIMINO, J. J., HUPP, J. A., AND HOFFER, E. P. Dxplain—An evolving diagnostic decision-support system. *J. Amer. Med. Assoc.* **258,** 67 (1987).
18. SALTON, G. Another look at automatic text-retrieval systems. *Commun. ACM* **29,** 648 (1986).
19. CRAIN, C. J. Protocol study of indexers at the National Library of Medicine. Appendix A of Carbonell, J. G., Evans, D. A., Scott D. S., Thomason, R. H., "Final Report on the Automated Classification Retrieval Project," Grant N01-LM-4-3529. National Library of Medicine, 1986.
20. SALTON, G. Historical Note: The past thirty years in information retrieval. *J. Amer. Soc. Inf. Sci.* **38,** 375 (1987).
21. SALTON, G. AND MCGILL, M. J. "Introduction to Modern Information Retrieval." McGraw-Hill, New York, 1983.
22. CARBONELL, J. G., EVANS, D. A., SCOTT, D. S., AND THOMASON, R. H. Final Report on the Automated Classification Retrieval Project," Grant N01-LM-4-3529. National Library of Medicine, 1986.
23. EVANS, D. A. "Final Report on the MedSORT-II Project: Developing and Managing Medical Thesauri." Technical Report No. CMU-LCL-87-3, Laboratory for Computational Linguistics, Carnegie Mellon University, 1987.
24. VRIES, J. K., SHOVAL, P., EVANS, D. A., MOOSSY, J., BANKS, G., AND LATCHAW, R. "An Expert System for Indexing and Retrieving Medical Information." University of Pittsburgh, 1986.
25. BORGMAN, C. L. Why are online catalogs so hard to use? Lessons learned from information retrieval studies. *J. Amer. Soc. Inf. Sci.* **37,** 387 (1986).
26. SALTON, G., FOX, E. A., AND WU, H. Extended Boolean information retrieval. *Commun. ACM* **26,** 1022 (1983).
27. HARMAN, D., BENSON, D., FITZPATRICK, L., HUNTZINGER, R., AND GOLDSTEIN, C. IRX: An information retrieval system for experimentation and user applications. *SIGIR Forum* **22,** 2 (1988).
28. RUBIN, R. H., Acquired Immunodeficiency Syndrome. *In* Rubenstein, R. *In* "Scientific American Medicine" (D. D. Federman, Ed.). Scientific American, New York, 1985.
29. RATHE, R., GREENES, R. A., COPE, L., AND GLASER, J. System architecture for a clinical workstation providing "point of use" knowledge access. *In* "Proceedings of the 13th Annual SCAMC." IEEE, New York, 1989, 669–672.
30. FRISSE, M. Searching for information in a hypertext medical handbook. *Commun. ACM* **31,** 880 (1988).
31. FUHR, N. Models for retrieval with probabilistic indexing. *Inf. Process. Manage.* **25,** 55 (1989).