ABSTRACT
The Expert/Expert-Locator (EEL) pair requests for technical information with appropriate technical organizations in a large research and development company. The system automatically constructs a semantic space of organizations and terms, using a statistical matrix decomposition technique (singular value decomposition) to represent semantic similarity present in large text sources [1]. In EEL, organizations are characterized by their documents. Using these documents as input, the analysis simultaneously fits organizations and the terms they use into the same 100-dimensional space. Similarity among organizations is determined by their overall pattern of term usage. Thus, organizations can be close to each other spatially without sharing the same vocabulary. Users' requests are processed and also fit into the high-dimensional space. The similarities between the request and all organizational objects in the space are computed, and the most similar organizations are returned to the user. We demonstrate that this technique is far superior to keyword matching used in most information retrieval systems.

1. INTRODUCTION
Current expert system technology can be applied quite reasonably to problems whose domains are self-contained and whose knowledge and rules can be accurately elicited from experts. It is often said that the bottleneck in expert system development is the knowledge acquisition phase. Indeed, it is extremely labor intensive and expensive to elicit knowledge and rules from even the best, most articulate experts. If the problem domain is extensive, such as codifying the knowledge present in a one volume encyclopedia [2], the undertaking consumes many person years. Even with the requisite labor to encode the knowledge, there are unavoidable problems of consistently coding the knowledge over time and over coders. Also, when such a task is completed, there is no guarantee that the structure of the knowledge imposed by the system's developers will correspond with the user's. Thus, the next significant hurdle for expert system technology is automatic representation of knowledge.

At the other end of the continuum from rule-based expert systems are standard information retrieval systems. Their scope of coverage is immense compared to most expert systems, often dealing with all the documents in a large library. These systems lack "rules" in the conventional artificial intelligence sense. Further, to say that they have "knowledge bases" is to mislead the term.

Most information retrieval systems use "keyword matching;" the user supplies a keyword or a set of keywords and the system returns those documents containing the user-specified keywords. While much is known about weighting keywords and preprocessing the keyword set in search and retrieval, the matching methods are crude with the result that the performance of such systems is poor. Many relevant documents in a collection are not retrieved, and many irrelevant documents are retrieved.

Efforts to improve keyword matching procedures include hand-crafted methods, such as: (a) constructing a thesaurus or a dictionary of synonyms, and (b) using a human intermediary to interpret the user's query. It is presumed that the human intermediary will know the keywords contained in the collection and will be able to rephrase the user's query into terms "known" by the system. However, there is notably poor agreement among expert intermediaries in the choice of search terms [3].

We now have a good empirical understanding of why keyword matching systems fail. The culprit is human language itself; human language is highly diverse. For any given concept or object there are many terms that are associated with it, so the chances of the user and the system agreeing on any one term are very small. In fact, Furnas, Landauer, Dumais and Gomez [4] found that the chances of two people choosing the same keyword for a familiar object was about 15%.

The number of computer accessible documents continues to increase at a rapid rate with much "knowledge" contained in these documents. Thus, it would be advantageous to devise methods for automatically representing information contained in large text sources without requiring explicit, expensive human expertise. Further, such methods should perform better than extant information retrieval methods.

We have developed a system that takes a request for technical information from a user and returns the most appropriate organizations in a technically diverse, 8,000 employee company. An application such as this is a rare feasible rule-based expert system project. There is no expert or experts; no small set of people understands in detail what each basic working group of five to ten people does. Most people's understanding of knowledge of a company's technical work is myopic; they know what their particular organization does, less about neighboring organizations, and even less about unrelated organizations. They have a "fish eye" view of the world [5]. Even if there were a set of experts, it is not at all clear how to encode the technical expertise of each organization in any known knowledge representation scheme.

2. DESCRIPTION OF SEMANTIC STRUCTURE ANALYSIS
The technique used here to represent company knowledge is "semantic structure analysis" [1]. In our application, organizations are represented by a representative collection of the technical documents they write. The large matrix (≈7100 terms by ≈500 organizations) of term frequencies within organizational documents is reduced to a large number of orthogonal dimensions (100) using singular value decomposition (see Section 2.2). Thus, the terms used by organizations
as well as the organizations themselves are placed in the same 100 dimensional space. Closeness of organizations is determined by their overall pattern of term usage, not by the particular terms or words used. Similarly, terms which are synonymous appear as neighbors in the space.

2.1. A Simple Example of the Method

Table 1 (adapted from [1]) will be used to illustrate how semantic structure analysis works. In this example, the universe consists of nine titles of technical papers with titles cl-c5 concerned with human/computer interaction and titles m1-m4 concerned with mathematical graph theory. If a user requested papers dealing with "human computer interaction," a keyword-based retrieval system would return titles cl, c2, and c4, since these titles contain at least one keyword from the user query. However, c3 and c5 while related to the query, would not be returned, since they share no words in common with the query.

| TABLE 1: SIMPLEx AMPLY XAMPLE OF SEMANTIC STRUCTURE ANALYSIS |
|-----------------------------|-------------------------------|
| **Title Set:**              |                               |
| cl: Human machine interface for Lab ABC computer applications |                               |
| c2: A survey of user opinion of computer system response time  |                               |
| c3: The EPS user interface management system                   |                               |
| c4: Systems and human systems engineering testing of EPS-2     |                               |
| c5: Relation of user-perceived response time to error measurement |                               |
| m1: The generation of random, binary, unordered trees          |                               |
| m2: The intersection graph of paths in trees                   |                               |
| m3: Graph minors IV: Widths of trees and well-quali- ordering  |                               |
| m4: Graph minors: A survey                                     |                               |

<table>
<thead>
<tr>
<th>TERMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
</tr>
<tr>
<td>c2</td>
</tr>
<tr>
<td>c3</td>
</tr>
<tr>
<td>c4</td>
</tr>
<tr>
<td>c5</td>
</tr>
<tr>
<td>m1</td>
</tr>
<tr>
<td>m2</td>
</tr>
<tr>
<td>m3</td>
</tr>
<tr>
<td>m4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TITLES</th>
</tr>
</thead>
<tbody>
<tr>
<td>human</td>
</tr>
<tr>
<td>interface</td>
</tr>
<tr>
<td>computer</td>
</tr>
<tr>
<td>system</td>
</tr>
<tr>
<td>response</td>
</tr>
<tr>
<td>time</td>
</tr>
<tr>
<td>EPS</td>
</tr>
<tr>
<td>survey</td>
</tr>
<tr>
<td>tree</td>
</tr>
<tr>
<td>graph</td>
</tr>
<tr>
<td>minor</td>
</tr>
</tbody>
</table>

**USER QUERY:**

"I want papers about human computer interaction."

In this example, the input to the semantic structure analysis is the term frequency of occurrence by title matrix. Since it is difficult to graphically represent more than a few dimensions, we show the term/title space in two dimensions. Figure 1 shows the results of the singular value decomposition in two dimensions. Note that titles and terms have been fit in the same space. First, the two types of titles form two distinct clusters; all the mathematical graph theory titles and terms occupy the same region of the space and are quite distinct from the cluster of human/computer interaction titles and terms.

To fit the query, "I want papers about human computer interaction" into the same two-dimensional space, those terms that intersect with terms in the title matrix are found, namely, "human" and "computer." To construct a query vector, the elements of the "human" and "computer" two-diagonal matrix (see below). In Figure 1, the query vector is designated by an arrow and a boxed "Q." Those titles and terms "close" to the query vector are the titles, cl-c5 and their associated terms. The measure of closeness or similarity is the cosine between the query vector and any given term or title vector. The cosine between the query vector and the cl-c5 terms and title vectors are all equal to or greater than 0.90. Thus, titles c3 and c5 would be returned as matches to the query, even though they share no common term; this is, of course, the desired result.
2.2. Brief Technical Description of Singular Value Decomposition.

Semantic structure analysis relies on a matrix decomposition technique called singular value decomposition. (For a more extensive technical description, see [6] [7].

Essentially, singular value decomposition can be used as a noise-reduction technique; the original matrix is represented in terms of a much smaller number of orthogonal dimensions. The number of dimensions required to represent adequately a particular domain is an empirical matter. If the number of dimensions is too large, random variation or noise in the input space will be modeled. If the number of dimensions is too small, significant semantic content will remain uncaptured.

We have found increasing accuracy in our system with 30, 50, and 100 dimensions. Thus, for our domain a solution of at least 100 dimensions is required.

Again, using the simple example, the term by document (or title) matrix, $Y$, can be decomposed into three other matrices: the document matrix (DOCUMENT), the term matrix (TERM), and a diagonal matrix of singular values (DIAGONAL):

$$Y_{td} = \text{TERM}_{tm} \text{DIAGONAL}_{mn} \text{DOCUMENT}_{nd}$$

where $Y$ is the original $t \times d$ matrix, TERM is a $t \times m$ column-orthogonal matrix with $m$ representing the dimensionality of the solution, and DOCUMENT$^T$ is the transpose of the column-orthogonal $d \times m$ DOCUMENT matrix. The $m \times m$ DIAGONAL matrix contains singular values in decreasing order of significance. The dimensionality of the solution ($m$) is the minimum of $t$ and $d$. For our purposes, we use $k$ less than $m$ singular values, where $k$ is empirically determined.

3. DESCRIPTION OF THE EXPERT/EXPERT-LOCATOR (EEL)

3.1. Document Processing

3.1.1. Text Selected

Each organization is represented by the collection of specially selected documents that describe it. For each of the company's first-level supervisory groups (five to ten people), we chose as exemplary documents the annual write-ups each must prepare (approximately 300 words), describing what work that group planned to undertake in the coming year. When possible, we supplemented each organization's project descriptions with the abstracts of their technical memoranda for the last 18 months. The total amount of text was approximately 3 megabytes.

3.1.2. Text Preprocessing

All text was preprocessed to isolate possible compound noun phrases. Rather than doing full-fledged parsing, which would have required too much computation for the amount of text, we performed "pseudo" or simple-minded parsing. Phrases were considered to be one of the following: all words between a list of (1) 161 stop words or (2) punctuation marks.

Since the semantic structure analysis is more effective the greater the number of terms included in the analysis, all inflectional suffixes (past tense, plurals, progressive tense, and adverbials) were removed from the words. Inflectional suffixes in contrast to derivational suffixes are thought not to change the meaning of the base word. Thus, stripping the "ion" from "information" changes the meaning, whereas removing the plural "s" from "network" does not change the meaning of the base word [8].

3.1.3. Creating the System Lexicon

3.1.3.1. Finding Compound Noun Phrases

The EEL lexicon contains many compound noun phrases. We have empirically demonstrated that knowing compounds substantially improves the system's performance. Even in the absence of empirical data, a gedanken experiment would suggest a large number of words occur in many semantically different environments. The more fully these environments can be represented, the richer the semantic space. For example, the "information" in "information retrieval" and "information theory" have different meanings. Representing "information retrieval" and "information theory" as compounds and treating each as separate terms in the semantic structure analysis ought to place each of these in different portions of the space. If a word occurs in radically different semantic environments, treating it as a single word in the semantic structure analysis will place the word close to the origin, whereas treating each of its different semantic environments separately should give better spatial differentiation.

Compound phrases were extracted using a semi-automatic procedure. First, phrases were found using pseudo-parsing procedures. Any phrase or subphrase that occurred in more than one document was a potential compound phrase. This list was then manually edited to include only noun phrases. Compound phrases ranged from two to eight words (e.g., "semi-insulating Fe-doped InP current blocking layer"). All compound noun phrases as well as the single words making up the compounds were entered into the semantic structure analysis. Of the 7,100 terms in the system lexicon, 2,879 were compounds.

3.1.3.2. Single Words

All inflectionally stripped single words that occurred in more than two organizations and that were not one of the 150 most frequent words in English were included in the system lexicon.

3.1.4. Semantic Structure Analysis

A singular value decomposition on 7,100 terms and 728 documents, representing 480 organizations in 100 dimensions was performed using PARAFAC [9] [10]. (Technical memoranda abstracts and work descriptions for a single organization were treated as two separate documents for the purposes of analysis.) Finding a 100 dimensional solution for a data matrix of this size is computer intensive even on a mini-super computer; however, this analysis is done only once.

3.1.5. Organizational and System Lexicon Databases

Each document in the system points to a single organization. The organizational titles corresponding to the documents were entered from a paper telephone directory into a hashed database. From an online company telephone directory, an organizational database was created containing the manager's name and mail address for each organization. Also, a database of each term in the system lexicon and its corresponding address in the semantic structure space was constructed.

3.2. Processing User Queries
3.2.1. Query Preprocessing

Inflectional suffixes in the user query are removed and potential phrases are identified.

3.2.2. Identify Terms in the System Lexicon

From the user’s query, the longest matching compound phrases as well as single words not part of compound phrases are found in the system lexicon.

3.2.3. Compute the Spatial Coordinates of the User Query

For each query term also contained in the system lexicon, the 100 dimensional vector is found. The query vector is the centroid of these 100 dimensional vectors.

3.2.4. Compute Similarities between the Query Vector and all Documents in the Space.

In order to guarantee that the best matching document is found, the query vector must be exhaustively compared to all documents in the space (here 728). The similarity metric used is the cosine or product moment correlation between the query vector and the document vectors. Thus, for each query, 728 cosines are computed. A cosine of 1.0 (a 0 degree angle) would indicate that the query vector and a particular document vector were on top of one another in the space.

3.2.5. Return the Best Matching Organizations to the User

The cosines are sorted and for each of the best N matching organizations (typically 8-10), the value of the cosine (ft), along with the organizational information are returned to the user. Table 2 shows sample input and output to EEL.

4. HOW WELL DOES THE SYSTEM WORK?

4.1. Comparison with Keyword Matching

If the semantic structure analysis is indeed capturing synonymy in its structure, it should perform better than standard keyword matching. This was tested directly with an early prototype of the system, whose lexicon consisted of 1800 single words and covered a smaller number of organizations (105). The documents that were used to form the semantic space were approximately 1500 technical memoranda abstracts. The semantic structure space was computed using 30 orthogonal dimensions.

The test queries were descriptions of current project assignments solicited from 40 people. A typical example is shown below:

**NAME OF PROJECT:**
automatic speech recognition

**SHORT DESCRIPTION:** Develop new distance metrics for ASR. Analysis of telephone speech database (to be collected by other Bellcore organizations) for creation of speaker independent templates for ASR.

**RESPONSIBILITIES:**
Design and run speech perception experiments (includes writing code for experimental paradigms). Consult with other Bellcore organizations on feasibility of ASR for automated billing of collect calls.

For each test query the person’s true organization was known. EEL and keyword matching were used to determine which of the 105 organizations were most similar to each project description. (For the keyword matching condition, terms were weighted by their inverse document frequency [11] If either method were perfect, the person’s true organization would be returned with rank 1, whereas chance performance would give a rank of 52.5. For the semantic structure analysis the median rank of the person’s true organization was returned with rank 5, whereas for the keyword matching condition, the rank of the true organization was 27.5. Clearly, semantic structure analysis out performed keyword matching or term overlap.

4.2. Using EEL to Predict the Similarity of New Technical Abstracts to Organizations

Using the current version of the system, we collected a set of 263 new technical abstracts that had not been used to construct the semantic space. Again, EEL was used to determine which organizations were most similar to each of the abstracts. EEL predicted the division (second level supervisory organization with approximately 20-30 people) with median rank 3. (A perfect score would median 1, and the chance ranking would be 52, since multiple documents refer to the same organization.) Overall, 75% of the distribution was contained within ranks 1 and 19. Entire abstracts are typically longer than queries users spontaneously generate. Thus, accuracy as a function of query

---

**TABLE 2**

**AN EXAMPLE OF A QUERY AND SYSTEM OUTPUT**

**QUERY:**
An Expert/Expert-Locating System Based on Automatic Representation of Semantic Structure.

**OUTPUT OF SYSTEM:**

**BEST ORGANIZATIONS**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name and Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>110.67</td>
</tr>
<tr>
<td>2</td>
<td>21233</td>
</tr>
<tr>
<td>3</td>
<td>21273</td>
</tr>
<tr>
<td>4</td>
<td>21326</td>
</tr>
<tr>
<td>5</td>
<td>21462</td>
</tr>
</tbody>
</table>

**ARTIFICIAL INTELLIGENCE AND INFORMATION SCIENCE RESEARCH GROUP**

**ARTIFICIAL INTELLIGENCE AND COMMUNICATIONS RESEARCH GROUP**

**COGNITIVE SCIENCE RESEARCH GROUP**

**EXPERIMENTAL SYSTEMS RESEARCH**

**SOFTWARE TECHNOLOGY**

Streeter, Lynn A : email to bellcore!lynn

Klein, T K : email to bellcore!tk

Riley, Christine A : email to thumper!car

Lewis, Carl P : email to ctlewis

---
length was tested using: (1) the title of each technical abstract, and (2) only a keyword or phrase from the title (selected by us). The median rank returned by the system of the true division using the title was 5 with 75% of the distribution within ranks 1 and 32, and for keyword or phrase from the title the median rank was 8 with 75% of the distribution between ranks 1 and 42.

The other important variable was number of dimensions used to represent the space. We tested 30, 50, and 100 dimensions, using the new abstracts as queries. For 30 dimensions, the median correct rank was 7 (75% of the distribution between ranks 1 and 32), and for 50 dimensions the median correct rank was 5 (75% of the distribution between ranks 1 and 23). Thus, best performance was obtained for 100 dimensions (median rank 3), and it is conceivable that a higher dimensionality would improve performance even more.

5. IMPLEMENTATION DETAILS
PROCESSING REQUIREMENTS AND CODE SIZE OF EEL:

1. Work Descriptions and Technical Memoranda abstracts:
   3 Megabytes of text.
2. Preprocessing of text for semantic structure analysis:
   1150 lines (C code).
3. Analysis program to compute semantic structure in 100 dimensions:
   4000 lines (Fortran).
4. Computation of 100 dimensional semantic structure space:
   A few days on a mini-supercomputer (Convex or Alliant).
   Analysis done once per information set in advance.

QUERY PROCESSING CODE:
1. Query processing program:
   2600 lines (C code).
2. Database routines:
   2000 lines (C code).
3. Program data (100 dimensional semantic space, term dictionary, organizational information, etc.):
   13 Megabytes.
4. Processing time per query:
   ~3 seconds on a Sun 3/75.

6. SUMMARY
EEL has proved quite useful for its intended purpose of pairing queries for information with the organizational expertise in a large, technically diverse company. In the process of demonstrating the system other equally appropriate applications have emerged somewhat serendipitously. These include: (1) which organizations visitors should see, given specific interests, (2) routing employment candidates' resumes, and (3) automatically determining dependencies among organizations for purposes of technology transfer.

The system's major strength is its ability to represent synonymy, terms that occur in similar contexts are spatially close to one another. There is however, no explicit mechanism for representing polysemy. A term that has several distinct meaning senses, such as "fly," will be represented as an amalgamation of its senses in the space. Incorporating knowledge of a large number of compound phrases is a step towards ameliorating this problem, but it is not a general solution.

Another shortcoming of the system is that EEL provides only an index of similarity or the degree of relationship among objects; there is no way to represent causation in the system. For instance, producers of a technology might like to know who their potential consumers are. In our system, a query is as likely to give the producer organizations for which it is a consumer as it is to give organizations for which it is a producer.

A minor problem is that the system does not yet have any mechanism to deal with negation. Thus, a query of the type, "I want Xs but not Ys," would give organizations most similar to the centroid of Xs and Ys, which is not the desired result. While conceptually this is a major problem, in actual practice no one has entered a query using a negation operator.

An additional limitation stems from the way in which the system is used. Users tend to underestimate the polysemy of language. For example, a user who entered "C" as a query was surprised to find a chemistry organization returned, since he had the C programming language in mind. In organizations with much technical diversity, acronyms tend to become overloaded; "AI" can mean "Artificial Intelligence" or "automatic intervention," depending on the context. Thus, the essential human limitation is to give too little context for a query, assuming that the system shares their internal semantic space. As has been shown, system performance increases steadily the longer the query. However, actual users tend to err on the side of brevity, but to the extent that these user predictions could be considered constants across all systems of this type, we have demonstrated that modeling the pattern of word usage in a high dimensional space produces better results than keyword matching and that such a structure can be derived by automatic means.
REFERENCES


