SIGIR 93
PP281-290

An Application of Least Squares Fit Mapping To Text Information Retrieval

Yiming Yang
Christopher G. Chute

Section of Medical Information Resources
Mayo Clinic/Foundation
Rochester, Minnesota 55905 USA

ABSTRACT

This paper describes a unique example-based mapping method for document retrieval. We discovered that the knowledge about relevance among queries and documents can be used to obtain empirical connections between query terms and the canonical concepts which are used for indexing the content of documents. These connections do not depend on whether there are shared terms among the queries and documents; therefore, they are especially effective for a mapping from queries to the documents where the concepts are relevant but the terms used by article authors happen to be different from the terms of database users. We employ a Linear Least Squares Fit (LLSF) technique to compute such connections from a collection of queries and documents where the relevance is assigned by humans, and then use these connections in the retrieval of documents where the relevance is unknown. We tested this method on both retrieval and indexing with a set of MEDLINE documents which has been used by other information retrieval systems for evaluations. The effectiveness of the LLSF mapping and the significant improvement over alternative approaches was evident in the tests.

1. INTRODUCTION

Current retrieval approaches can be grouped into two major categories: surface-based retrieval (using words in texts) and concept-based retrieval (using canonical indexing terms). The former determines the relevance based on whether the terms in a query match the terms in a document. As it is simple and therefore commonly used, surface-based retrieval methods share a significant weakness in that they ignore the information within non-shared terms; poor retrieval of relevant documents is unavoidable due to the fact that relevant contents are often represented in a variety of terms. Concept-based retrieval first identifies the contents in queries and documents using canonical concepts (indexing terms), and then determines the relevance based on whether the concepts in a query match the concepts in a document. In most practical databases, documents have been indexed by human experts and the difficult part of retrieval is to convert user queries into the right indexing terms. Naive users who are not familiar with the indexing terms of a particular database cannot avoid a poor retrieval, and even experienced users (e.g. librarians) do not always guarantee a sufficient or consistent keyword assignment to queries [1].

Currently, considerable research effort has been given to the development and use of terminology thesauri or knowledge bases, in order to improve the mapping from queries to canonical concepts [2], [3] [4]. A major problem with using thesauri is the difficulty of finding a thesaurus which satisfies the specific needs of applications. Often in thesaurus development there are no clear principles about what entries or relationships should be included, what should be used for particular applications, and how to prevent ad hoc decisions by humans. As Salton pointed out, there is "no guarantee that a thesaurus tailored to a particular text collection can be usefully adapted to another collection. As a result, it has not been possible to obtain reliable improvements in retrieval effectiveness by using thesauruses with a variety of different document collections" [5].

Salton's solution for improving a retrieval is to expand or refine queries using relevant documents instead of using thesauri. The method, known as "relevance feedback" [6], requires a user interaction in the retrieval process to identify some relevant documents for each query. The words in the relevant documents then are weighted and added to the initial query, and the expanded query is used for further retrieval of relevant documents. The underlying idea is that the words in a relevant document are likely to be relevant to the query and thus may be helpful for finding relevant documents which cannot be found using the original query words alone. Such an assumption, however, is rather weak because many words in a relevant document could be irrelevant to the query. While the expansion of a query increases the chance of finding more relevant documents, the chance

of ending up with irrelevant documents also increases; thus the gain in recall (refer to Section 3.2 for definition) generally costs as a loss of precision. How can such a tradeoff be controlled? How can an ambiguity explosion be avoided when expanding queries? These are essential questions for the effectiveness of retrieval methods using relevance information, but no satisfactory answers have been found. Another limitation of the feedback scheme is the need for relevance information of every query. It would be desirable that the search not depend on user interaction, and that previous retrieval experiences be used for predicting answers of new queries which are not necessarily the same as the previous queries.

This paper introduces a unique example-based mapping method for text retrieval. We have introduced such a method for text classification [7] [8], and here we focus on its features as a retrieval method. The mapping from queries to documents is not based on surface matching. We share the part of concept-based retrieval of using indexing terms, but do not require terminology thesauri for the mapping from queries to indexing terms. We achieve the functionality of thesauri by algorithmically "learning" empirical connections between query terms and indexing terms from a set of relevant documents assigned by humans. Weights on these connections are calculated in a way that the context constraints in the matched queries and documents (the "training set") are preserved, and these constraints minimize the ambiguities in mapping an arbitrary query to documents. A Linear Least Squares Fit (LLSF) technique enabled us to obtain such a mapping function and predict a most likely match (a document) or a set of most likely matches for an arbitrary query, based on the likelihood suggested in a training set. Our method does not require queries for retrieval to be the same as the ones in the training set.

2. THE METHOD

2.1 The Mapping Between Two Vector Spaces

The goal is to find a mapping from an arbitrary query to a set of relevant documents, and we are interested in a concept-based approach. Assuming that all documents are already indexed using canonical concepts, our focus is the mapping from queries to canonical concepts. In our model, a query is treated as a set of weighted terms, ignoring word order, and a document as a set of weighted concepts. We use "term" to simply mean a word, and "concept" to mean the conceptual unit in an indexing language, which is typically an indexing phrase (we will also discuss the use of document words as a special case of conceptual unit in a later section, but in most of the paper we focus on canonical indexing phrases used in existing databases). We will use "concept" instead of "indexing terms" below, because our approach treats an indexing phrase as merely an atomic identifier of a subject category, which can be substituted by any other identifier without effecting the mapping process.

The basic idea is to find likely connections between query terms and canonical concepts, according to a given set of matched queries and documents. In mathematics, there are well established methods for computing unknown points from known points, for example, the interpolation techniques. Similarly, we want to find the unknown matches from the known matches. In order to achieve such a goal, we need a numerical model for the query-to-document mapping. We define a mapping between two multidimensional vector spaces, from the source space to the target space. The source space is a lexical space with unique terms as the dimensions. The target space is a conceptual space with unique concepts as the dimensions. A query is a point (a vector) in the source space and a document is a point in the target space. Our approach is to find a mapping function to project a query in the source space to its image in the target space, and then rank the relevance of documents according to their closeness to the image of the query. A training set is needed for obtaining such a mapping function.

2.2 The Training Set

The training set we need is a set of queries for which both a term based lexical expression and a conceptual expression are given. Such a training set can be derived from a set of matched queries and documents. We use the terms in the given queries for the dimensions of the source space, and the concepts in the relevant documents for the dimensions of the target space. Let D be a document collection for which the relevance to the query q needed to be determined (these documents may or may not have an overlap with the documents used for training). A query in the source space is represented as a vector of term weights. We apply a commonly used statistic weighting scheme which combines term frequency and an inverse document frequency (IDF) [5] to our model as described below.

A term weight contains two factors, the first of which is the inverse document frequency of term t with respect to (w.r.t.) the document collection D, represented in

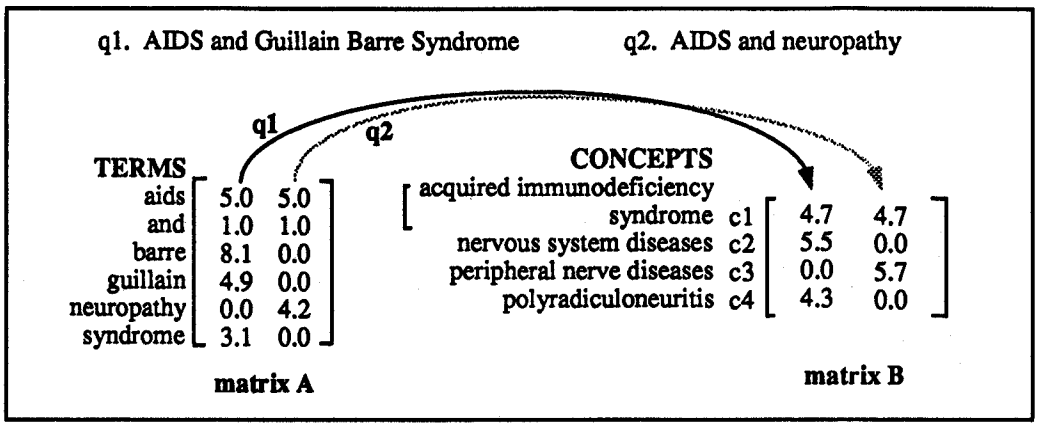

Figure 1. The query vectors in the source space and the target space

terms:

$$\text{IDF}_t\ (t, D) = \log\left(\frac{\text{the number of documents in } D}{\text{the number of documents with term } t}\right) + 1.$$

The second factor is the frequency of term t in query q:

$$\text{TF}\ (t, q) = \text{number of times term } t \text{ occurs in query } q.$$

The weight of term t in query q w.r.t. collection D is

$$\text{TW}\ (t, q, D) = \text{IDF}_t\ (t, D) \times \text{TF}\ (t, q).$$

A document in the target space is represented as a vector of concept weights. The weight of a concept in a document (d) also contains two factors. The first factor is the inverse document frequency of concept c w.r.t. document collection D (represented in concepts):

$$\text{IDF}_c\ (c, D) = \log\left(\frac{\text{the number of documents in } D}{\text{the number of documents with concept } c}\right) + 1.$$

The second factor is the frequency of concept c in document d:

$$\text{CF}\ (c, d) = \text{frequency of concept } c \text{ in document } d.$$

The weight of concept c in document d w.r.t. document collection D is

$$\text{CW}\ (c, d, D) = \text{IDF}_c\ (c, D) \times \text{CF}\ (c, d).$$

The conceptual representation of a training query is a vector of concept weights derived from the relevant docoments:

$$\text{QCW}\ (c, D_q, D) = (\text{frequency of concept } c \text{ in } D_q) \times \text{IDF}_c\ (c, D)$$

where D_q is the set of documents relevant to query q.

Figure 1 shows the matrix representation of a training set containing two queries. Matrix A represents the lexical expressions of the queries, where a row corresponds to a term, a column represents a query, and a cell contains the weight of a term in the corresponding query. Matrix B represents the conceptual expressions of the queries. A row of B is a concept, a column represents the same query in the corresponding column of A, a cell contains the weight of a concept in the corresponding query.

2.3 The LLSF Mapping Function

Having matrices A and B, we are ready to compute the mapping function. The mapping function is a transformation matrix (denoted as W) from the source vector space to the target vector space as defined below in the LLSF problem.

Definition 1. The LLSF problem is to find W which minimizes the sum

$$\sum_{i=1}^{k} \|\vec{e}_i\|_2^2 = \sum_{i=1}^{k} \|W\vec{a}_i - \vec{b}_i\|_2^2 = \|WA - B\|_F^2,$$

where $\vec{a}_i$ is an $n \times 1$ source vector, $\vec{b}_i$ is an $m \times 1$ target vector, $A_{n\times k} = [\vec{a}_1, \vec{a}_2, ..., \vec{a}_k]$, $B_{m\times k} = [\vec{b}_1, \vec{b}_2, ..., \vec{b}_k]$, $\vec{a}_i$ and $\vec{b}_i$ are a matched pair, $\vec{e}_i \stackrel{\text{def}}{=} W\vec{a}_i - \vec{b}_i$ is the mapping error of the ith pair,

$$\|\ldots\|_2 \stackrel{\text{def}}{=} \sqrt{\sum_{j=1}^{m} v_j^2}$$

is vector 2-norm of an $m \times 1$ vector, and

$$\|\ldots\|_F \stackrel{\text{def}}{=} \sqrt{\sum_{i=1}^{k}\sum_{j=1}^{m} m_{ij}^2}$$

CONCEPT	TERM					
	aids	and	barre	guillain	neuropathy	syndrome
acquired immunodeficiency syndrome	.57	.12	.14	.08	.41	.05
nervous system diseases	.10	.02	.41	.25	-.13	.16
peripheral nerve diseases	.58	.12	-.25	-.15	.62	-.10
polyradiculoneuritis	.17	.04	.69	.42	-.21	.27

Figure 2. An LLSF solution W of the linear system WA = B

is the Frobenius matrix norm of an $m \times k$ matrix.

Theoretically, the LLSF problem always has at least one solution. A conventional method for solving the LLSF is to use singular value decomposition (SVD) [9] [10]. Since mathematics is not the focus of this paper, we simply outline the computation without proof.

Given matrix A $(n \times k)$ and B $(m \times k)$, the computation of an LLSF for $WA = B$ consists of the following steps:

(1) Compute an SVD of A, yielding matrices U, S and V:

if $n \geq k$, decompose A such that $A = USV^T$,

if $n < k$, decompose the transpose A^T such that $A^T = VSU^T$,

where U $(n \times p)$ and V $(k \times p)$ contain the left and right singular vectors, respectively, and V^T is the transpose of V; S is a diagonal $(p \times p)$ which contains p non-zero singular values $s_1 \geq s_2 \geq ... \geq s_p > 0$ and $p \leq min(k, n)$;

(2) Compute the mapping function $W = BVS^{-1}U^T$, where $S^{-1} = diag(1/s_1, 1/s_2, ..., 1/s_p)$.

This algorithm has a time complexity of $O(nk^2)$ where k is the number of pairs in the training set and n is the number of distinct terms in the source space $(k > n)$. Figure 2 shows an LLSF solution of the training set in Figure 1.

2.4 The Term-to-Concept Connections

The matrix W has an intuitive meaning in mapping a source vector to a target vector: it is equivalent to a set of connections between source terms and target concepts. Note the difference between the weights here and the term weight or the concept weight mentioned in the previous section. The term weight means how important a term is counted in a query, and it depends on the term frequency in the query and the term distribution over the document collection. Concept weight is similar. The weight on a term-to-concept connection means how likely a term is related to a concept. It depends on the co-occurrence of the term and the concept in the training set, and on the distribution of the co-occurrences of other terms and concepts in the training set. The LLSF assigns weights to the connections in such a way as to globally minimize the mapping errors for the pairs of queries and documents in the training set. A more informative term has weighted connections biased toward some particular concepts, while a less informative word has relatively even weights on the connections toward all the concepts. If a term has never co-occurred with a concept in the training set, the weight on the connection is zero, no matter how much the term weights or the concept weights.

Not correct. Refer to TOIS 94 Yang's paper for the correction

These connections enable the system to "interpret" an arbitrary query into a set of weighted concepts. Figure 3 shows how a source vector is mapped to a target vector. Given a query *What is the mechanism of Guillain Barre Syndrome?*, we form a vector $\vec{x}$ =(0 0 8.1 4.9 0 3.1) in the source space (terms not included in the training set are ignored). Vector $\vec{x}$ is transformed into a target vector $\vec{y} = W\vec{x}$ =(1.7 5.0 -3.0 8.4). Vector $\vec{y}$ is the image of the query $\vec{x}$, which suggests that the query is most likely (with a weight of 8.4) to be relevant to concept *polyradiculoneuritis*, more or less relevant to *nervous system diseases* (with a weight of 5.0) and *acquired immunodeficiency syndrome* (with a weight of 1.7), but most unlikely (with a weight of -3.0) to be a match of concept *peripheral nerve diseases*. Of course, this is an over-simplified example; in real use, the training set is much larger and the adjustment of weights is affected by term/concept distribution over thousands of query/document pairs.

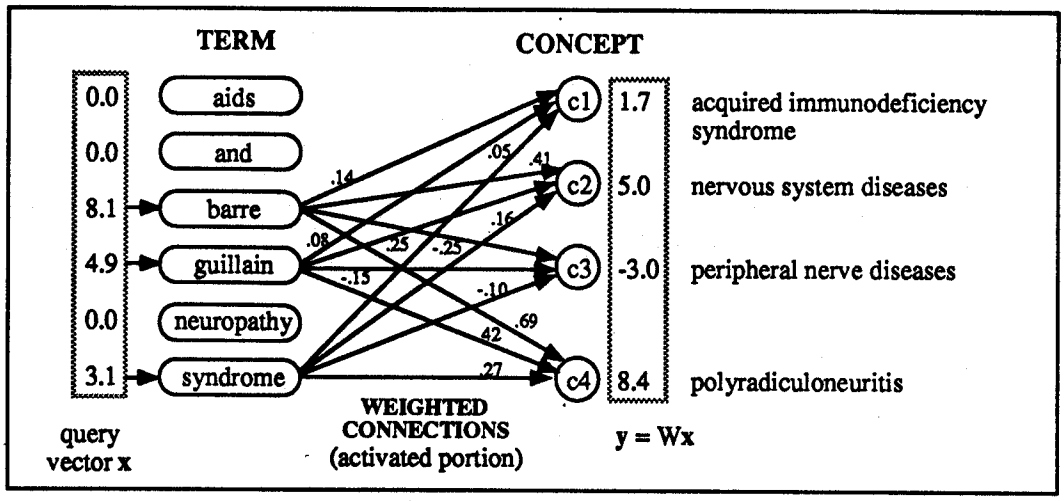

Figure 3. An example of the LLSF mapping via term-to-concept connections

2.5 The Relevance Score of Documents

The last step of the mapping process is to rank the documents as the candidates matching a particular query. For ranking the relevance we use the cosine value, a common measure of vector similarity in linear algebra and also a measure widely used by information retrieval systems for comparing vectorized texts. We define the relevance score of a document with respect to a query below.

Definition 2. The relevance score of document $\vec{d}$ with respect to query $\vec{x}$ is

$$\text{relevance(query } \vec{x}, \text{document } \vec{d}) = \cos(\vec{y}, \vec{d}) = \frac{y_1 d_1 + y_2 d_2 + ... + y_m d_m}{\sqrt{y_1^2 + y_2^2 + ... + y_m^2}\sqrt{v1_1^2 + d_2^2 + ... + d_m^2}}$$

where $\vec{x} = (x_1, x_2, ..., x_n)$ is a query vector in the source space, $\vec{y} = W\vec{x} = (y_1, y_2, ..., y_m)$ is the image of $\vec{x}$ in the target space, and $\vec{d} = (d_1, d_2, ..., d_m)$ is a document vector in the target space.

Using this measure, we can obtain a ranked list of documents for any query. The relevance score ranges from -1 to 1, and the document with the highest score is considered to be the most likely answer.

3. THE TESTS

3.1 The Testing Data

We used a testing set of MEDLINE retrievals for our evaluation. MEDLINE is one of the world's largest and most frequently used online databases. A testing set was designed by Haynes and McKibbon for a recent evaluation of MEDLINE [1]. The set consists of 78 queries and 3,403 citations. A citation is a data entry of MEDLINE, each containing a title and/or an abstract, and a set of subject categories (Medical Subject Headings, or MeSH) [11] assigned by human experts from the National Library of Medicine. Adapting our terminology to this, we call the title and the abstract together a document, and the subject categories the concepts. Roughly half of the documents are relevant to the queries. Three versions of the queries were tested in the MEDLINE evaluation. The original queries were written by novice users (physicians and medical students) who were not familiar with using MEDLINE. These queries are rewritten by physicians expert in using MEDLINE. Another refined version of the queries was given by librarians. The relevance between queries and documents was assigned by a clinician who was expert in the area of the search topic. In the later comparisons, we refer to the MEDLINE tests as Novice, Expert and Librarian, according to the different versions of the queries.

The original testing set (the Novice version) was later used by Hersh for an evaluation of SAPHIRE [12], a retrieval system using a large terminology thesaurus [2] for mapping queries and documents into canonical concepts. Hersh reduced the testing set by eliminating documents in which an abstract was not present and queries which did not have relevant documents in the reduced document collection. The resulting set has 75 queries (the Novice queries) and 2,344 documents. We will refer to the reduced testing set as the Shared Testing, because we use this set for our test.

Table 1. The recalls and precisions of the LLSF mapping and SMART

RECALL (percent)	PRECISION (percent)			
	on the Disjoint Testing set			on the Shared Testing set
	LLSF	SMART+	SMART-	SMART-
10	69	69	60	64
20	67	67	56	61
30	65	64	50	56
40	61	63	49	54
50	59	60	47	51
60	53	52	42	50
70	48	45	33	46
80	42	32	29	40
90	39	39	25	32
100	33	30	22	26

3.2 The Primary Test

The 2,344 documents in the Shared Testing set contain 991 documents (42%) which are relevant to the query set and 1,353 documents (58%) which are irrelevant to any of these queries. We split the data into a training set and a testing set. We sorted the relevant query/document pairs (1,074) by document, and took the documents in the odd pairs for training, and the other documents for testing. The resulting training set contains 71 queries and 524 relevant documents, and the testing set contains 68 queries and 1,820 documents. Only 22% of the testing documents are relevant to the testing queries and 78% are irrelevant. Eighty-eight percent of the testing queries are contained in the training set, but there is no overlap among the training documents and the testing documents. We call the training set "Disjoint Training" and the testing set "Disjoint Testing". We split the data this way so that most of the testing queries can use the term-to-concept connections obtained from the training set, but none of the known answers is included in the testing set. The test is to verify how much the LLSF mapping can capture unknown matches (documents) after training from the knowns. An alternative choice is to use the Shared Testing set instead of the Disjoint Testing set for the evaluation. The result on the Shared Testing set would be better because the training documents are contained in the testing set. However, it would be unfair if we compare such a result with MEDLINE and SAPHIRE since the testing conditions would be different: for LLSF, there would be roughly 50% of the relevant documents already known before the retrieval; for MEDLINE and SAPHIRE, 0% was known. For a fair comparison, we exclude the training documents in the testing set.

The result of the LLSF mapping on the Disjoint Testing set is shown in Table 1. We use the conventional measures, recall and precision, for evaluating the retrieval results.

Definition 9. The recall and precision of a retrieval with respect to query q are

$$\text{recall } (q) = \frac{\text{the number of documents retrieved and relevant to } q}{\text{the total number of documents relevant to } q}$$

$$\text{precision } (q) = \frac{\text{the number of documents retrieved and relevant to } q}{\text{the total number of documents retrieved}}$$

For a set of queries, we compute the recall and precision for each query and then average them: for recall threshold at 10%, 20%, 30% ... 100%, retrieve as many documents as needed for each query, and average the precisions of the points where the threshold is achieved.

In the test, a preprocessing was applied on the queries and indexing terms of documents to remove punctuation and numbers and to change uppercase letters to lowercase; no stemming or removal of common terms was applied. We did not use human editing or modification on the testing data. Our experimental system is implemented as a combination of C++, Perl and UNIX shell programming. For singular value decomposition (SVD), currently we use a matrix library in C++ [13] which implements the same algorithm as in LINPACK[14]. The LLSF computation on the Full Training set (75 queries represented in a source space with dimensions of 310 terms and a target space with dimensions of 2,281 concepts) took 67 seconds in total on a SUN SPARCstation 2, including 12 seconds for the SVD computation. The retrieval took 0.81 seconds per query. This performance is satisfactory for practical needs.

3.3 The Comparison

Since two different testing sets were used, we cannot make a direct comparison of the LLSF result with MEDLINE and SAPHIRE. So we ran the SMART system on both sets for an indirect comparison. The SMART system is developed by Salton's group [5] [6] and is recognized as one of the most representative retrieval systems. It has been used for comparison in evaluations of retrieval systems including SAPHIRE.

We tested SMART in two cases: without and with using relevance information. We distinguish these two cases by SMART- and SMART+. The former tested SMART on the original queries (the MEDLINE Novice version), and the latter tested SMART on the expanded queries using relevant documents in the Disjoint Training set. We ran SMART with the default setting of its parameters in the statistic term weighting scheme which combines term frequency and an Inverse Document Frequency (IDF). We did not apply the relevance feedback part of SMART because that would use different relevant documents for query expansion, and this would be inconvenient for the comparison with the LLSF method. Since our focus is on the effectiveness of relevance information and not on the user interaction part of SMART, this modification would not be inappropriate. The results of SMART- and SMART+ are also included in Table 1.

Figure 4 shows the retrieval results on the Shared Testing set under the condition that the relevance information was not used. The methods include MEDLINE, SAPHIRE (according to the published data [3] [12]) and SMART-. Figure 5 shows the retrieval results on the Disjoint Testing set. The methods being compared are SMART+, LLSF and SMART-. Combining Figures 4 and 5, we can make some observations about the problems and the effectiveness of the approaches.

MEDLINE's results on the Shared Testing set had recall and precision of (42.3, 39.7) for novice users, (51.3, 46.6) for expert users, and (52.6, 59.8) for librarians. This indicates that the mapping of a query to the appropriate indexing terms is crucial in a database retrieval where documents are indexed by canonical terms.

SAPHIRE, employing a thesaurus [2] which is especially tailored to the canonical terms used in MEDLINE and which has a very large collection of synonyms (78,244), did not improve on the retrieval by much. SAPHIRE had a small improvement over MEDLINE for novice users, but was worse than MEDLINE for expert users and librarians.

SMART- had a performance better than MEDLINE for expert users but worse than MEDLINE for librarians. The interesting point is that the use of a thesaurus in SAPHIRE did not result in any improvement over SMART- which only used surface words in the queries and documents. This is another example showing the difficulty of finding a thesaurus satisfying the specific needs of applications.

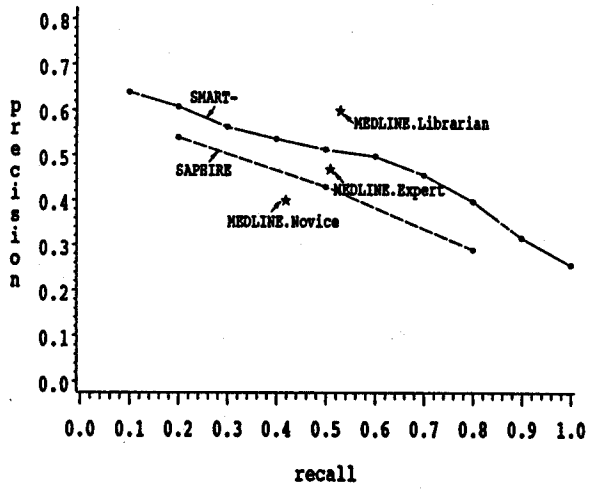

Figure 4. The different methods without using relevance in document retrieval on the Shared Testing set

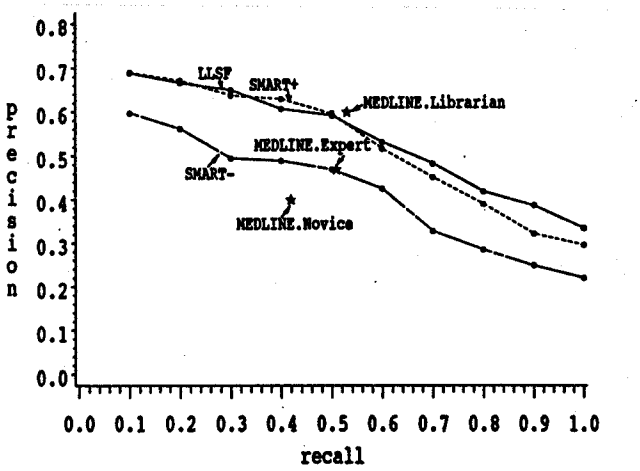

Figure 5. The different methods in document retrieval on the Disjoint Testing set

The LLSF mapping and SMART+ had very similar achievements in the retrieval and both have a significant improvement over SMART-. This suggests that connecting/expanding the original queries to broader sets of concepts/words was a major reason for the improvement, and the LLSF method and SMART+ were equally effective at this. There is a common feature between these two methods, that is, they both use empirical connections between queries and documents,

and these connections come from humans' knowledge about the relevance. In this sense, they are both concept-based (or partly concept-based) approaches.

We would like to point out a difference between the two testing sets. Comparing the recalls and precisions of SMART- on the two testing sets in Table 1, we can say that the Disjoint Testing set is more difficult than the Shared Testing set (partly because there are more irrelevant documents in the Disjoint Testing set, viz. 78% versus 58%). Taking this into account in the comparison, we can say that both LLSF and SMART+ achieved the level of MEDLINE for librarians.

4. OTHER ASPECTS

4.1 Handling Ambiguities

We have evaluated the effectiveness of the LLSF method in document retrieval. A closely related problem is document indexing. Retrieval and indexing are two different tasks. Retrieval requires a mapping from a query to the relevant documents. Indexing requires a mapping from a document to the relevant indexing terms. The LLSF model can show their similarity, since they both require a mapping from text to relevant concepts. As a further evaluation of the LLSF method, we tested the indexing aspect with the MEDLINE data.

We use the same testing sets as we used to test the document retrieval: Disjoint Training set for training and the Disjoint Testing set for evaluation. The training set contains 7,187 document/concept pairs. Documents in the indexing process correspond to the queries in the retrieval process. So a difference between the indexing test and the retrieval test is that the testing "queries" (documents) are a disjoint set of the training "queries" in this test. For comparison, we also tested SMART- and SMART+ for indexing. Since there are no overlap in the documents, there would be no use of expanding the "queries" (documents) as we did in document retrieval with the SMART+ test. We then expanded the expression of target items (concepts) instead, that is, adding words in the relevant documents to the corresponding indexing phrase.

Figure 6 shows the results of these methods. The LLSF had significant improvement over SMART- which was much better than SMART+, except at the low precision end. This is different from our experience with the retrieval tests where SMART+ was almost equally good as the LLSF. Obviously, a big loss of precision was paid by SMART+ as the price for the recall. The high precision end in fact is more important in real word applications, because only a few of the top-ranking candidates would be brought to user's attention when we use the automated mapping to assist human indexing.

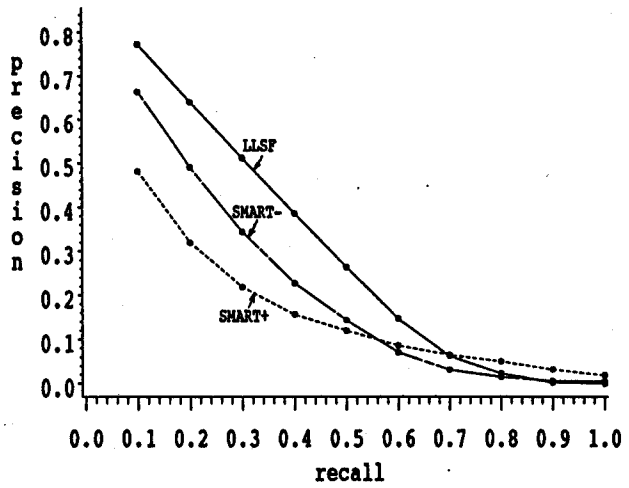

Figure 6. The different methods tested on document indexing

What caused the poor performance of SMART+ in indexing? It is worthwhile to analyze the difference between the training data used for the indexing test and the retrieval test. The same document set (Disjoint Training) was used, however, the relevance information is more ambiguous for indexing. There are about 14 canonical concepts per document in average which means that only a small portion of a relevant document is actually relevant to a concept. By adding the words in all the relevant documents to an indexing phrase and then using the expanded expression of a concept during the mapping process, an ambiguity explosion is imaginable. The LLSF mapping, on the other hand, has the functionality to impose the context constraints in the mapping, that is, the concepts of the training document(s) which are most close to a testing document will be chosen as the most likely candidates. The context constraints effectively reduced the ambiguities to the minimum.

The result of the LLSF method in this test is much worse than those we obtained in other classification tests with clinical data. In those tests, the relevance information was more precise, e.g. one category per diagnosis or four categories per case in average; the results were typically 89% precision at 100% recall. This comparison suggests that document indexing could be significantly improved if better training data were available. Nevertheless, this test gave a quantitative observation on how much we could achieve from the currently available training data (millions of indexed

documents are available in many databases), and interesting insights on the power of different methods in handling noise or ambiguities. Note that the results of SMART+ in the indexing test should not be interpreted as the performance of Salton's relevance feedback scheme because the latter is only designed for the case that relevance information is available for each query, and does not apply to the queries which are different from the training set.

4.2 Using Words instead of Concepts

We have been using canonical concepts in the explanation of our approach, but *Concept* is not a necessary choice for the target dimension in our model, an alternative choice is to use *term*. That is, we can use terms to represent documents, instead of using concepts. Such a use has been described in previous papers [7] [15].

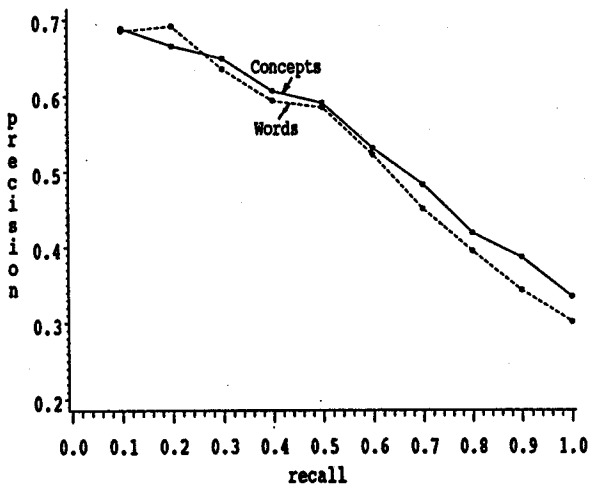

Figure 7. Using concepts versus words in document retrieval

Figure 7 shows the results of the two alternative choices tested on the Disjoint sets; using original terms in the documents did not make any significant difference than using the MEDLINE canonical concepts (MeSH terms). The major reason for the achievement of the LLSF mapping comes from the power of the method, and not from a particular choice of the target dimensions. We call our approach "concept-based" in the sense of a conceptual mapping from source queries to target documents, but not in the sense of using canonical concepts as indexing units in the document space. Another reason of focusing on the use of canonical concepts in this paper is that most practical databases use canonical concepts for organizing the data or for indexing the contents.

From a computational point of view, there is a difference between using terms and concepts. For example, the Shared Testing set contains 16,201 unique words and 4,020 unique concepts; using words would lead to a problem 16 times larger, a quadratic increase based on the current algorithm. However, *term* would be a good choice if the computation is affordable, or if canonical terms are not available or not suitable for a particular set of queries.

4.3 Further Research

The key point of our method is its *example-based* nature. We do not think the LLSF is the only choice for the computation of our model. Fitting techniques other than the LLSF are possible. The fact that the LLSF mapping has well known mathematical properties makes it preferable at this stage of research. While the LLSF is simpler and faster to compute than neural networks, it is still relatively expensive and impractical at the current stage for very large training sets (the largest training set we successfully tested contained 7,276 source terms and 1,610 target concepts). Seeking more efficient numerical solutions for the LLSF, or alternative statistical methods for capturing term/concept correlations, remain our ongoing research topics.

5. SUMMARY

A unique example-based approach to automated document retrieval and indexing is introduced. We discovered that the knowledge about relevance among queries and documents can play a key role in capturing the underlying concepts in user queries, and such knowledge can be "learned" from empirical data. We use a set of relevant queries and documents for training, and employ a Linear Least Squares Fit (LLSF) technique to compute a likely mapping from an arbitrary query to a set of documents, according to how terms in queries and concepts in documents are correlated (or co-occur) in the training set. The empirical term-to-concept connections captured by the LLSF achieve the functionality of terminology thesauri, and this makes our approach especially effective for a mapping from queries to the documents which are conceptually relevant but do not necessarily have any surface matching. The LLSF mapping also preserves the context constraints present in the training data, and therefore is more powerful in handling ambiguities compared to other mapping methods that do not use context constraints. These features of the LLSF mapping were tested with the MEDLINE testing data and the effectiveness was evident.

Acknowledgement

We would like to thank William R. Hersh for generously providing the testing data and Geoffrey Atkin for programming. This work is supported in part by NIH support grants LM-07041, LM-05416, and AR30582.

References

1. Haynes R, McKibbon K, Walker C, Ryan N, Fitzgerald D, Ramsden M. Online access to MEDLINE in clinical settings. *Ann. Int. Med.* 1990;112:78–84.

2. Lindberg D, Humphreys B. The UMLS knowledge sources: tools for building better user interfaces. *Proc 14th Ann Symp Comp Applic Med Care* 1990;14:121–125.

3. Hersh WR, Haynes RB. Evaluation of SAPHIRE: an automated approach to indexing and retrieving medical literature. *Proc 15th Ann Symp Comp Applic Med Care* 1991;15:808–812.

4. Cousins SB, Silverstein JC, Frisse ME. Query networks for medical information retrieval - assigning probabilistic relationships. *Proc 14th Ann Symp Comp Applic Med Care* 1990;14:121–125.

5. Salton G. Development in Automatic Text Retrieval, *Science* 1991:253:974-980.

6. Wu H, Salton G. The estimation of term relevance weights using relevance feedback. *J Documentation* 1981;37:194-219.

7. Yang Y, Chute CG. A Linear Least Squares Fit mapping method for information retrieval from natural language texts. *Proc 14th International Conference on Computational Linguistics* 1992:447–453.

8. Yang Y, Chute CG. An application of least squares fit mapping to clinical classification. *Proc 16th Ann Symp Comp Applic Med Care* 1992;16:460–464.

9. Lawson CL, and Hanson RJ. *Solving Least Squares Problems.* Englewood Cliffs, N.J.: Prentice-Hall, 1974.

10. Golub GH, Van Loan CE. *Matrix Computations, 2nd Edition.* Baltimore, MD: The Johns Hopkins University Press, 1989.

11. *Medical Subject Headings (MeSH).* Bethesda, MD: National Library of Medicine, 1993.

12. Hersh WR, Hickam DH, Leone TJ. Words, concepts, or both: optimal indexing units for automated information retrieval. *Proc 16th Ann Symp Comp Applic Med Care* 1992;16:644–648.

13. *M++ Class Library, User Guide, Release 3.* Bellevue, WA: Dyad Software Corporation, 1991.

14. Dongarra JJ, Moler CB, Bunch JR, Stewart GW. *LINPACK Users' Guide.* Philadelphia, PA: SIAM, 1979.

15. Yang Y, Chute CG. A numerical solution for text information retrieval and its application in patient data classification. *Technical Report Series, No. 50,* Section of Biostatistics, Mayo Clinic, Rochester, MN, 1992.