



Context Vector-Based Text Retrieval

A Fair Isaac White Paper

William R. Caid

Joel L. Carleton

August 2003

Table of Contents

Introduction.....	2
Background	3
The Context Vector Concept.....	5
Word Context Vectors	6
Assessment of Learned Relationships.....	8
Generation of Document Context Vectors	10
Document Retrieval.....	11
Ad Hoc Retrieval and Routing Performance	12
A. Training Stem Vectors	12
B. Ad Hoc Retrieval Experiments	12
C. Routing Experiments	13
Document Clustering and Cluster Trees	15
Document Classifier Determination and Automatic Classification	16
Conclusions and Comments	17
References	18
About Fair Isaac	19

Introduction

Fair Isaac has developed a neural network-based approach to learning word associations and usages based upon training on large volumes of free text. This approach is based upon the concept of “context vectors” and is embodied in a system called MatchPlus. A context vector is associated with every unique word in the training corpus. A self organization-based learning approach is used to derive these context vectors such that vectors for words that are used in similar contexts will point in similar directions. Unlike other vector space techniques that associate vectors with words, the MatchPlus approach exploits local rather than document-wide context for learning these similarity of usage relationships. As a result, the MatchPlus approach can learn word usages and can perform word sense disambiguation.

Once word context vectors have been learned, context vectors for documents and queries can be computed as a weighted sum of word context vectors. Document retrieval is performed based on Euclidean distance between a query context vector and document context vectors. Document context vectors can be clustered to form self organized subject indices. Index contents can be identified (summarized) by determining word vectors that are close to cluster centers.

Background

The traditional approach to text retrieval is based on the concept of Boolean queries or “word matches”. These systems allow the user to formulate queries in terms of the existence of one or more “key words” contained in documents. Identification of these key words provides a basis for document retrieval. In practice these Boolean match systems return far too many documents for the user to effectively review. The information overload forces the user to suggest additional query terms to qualify and reduce the number of matching documents. However, this increases the risk that the actual documents of interest may not contain the key word(s) as specified in the user query. This results in important documents not being retrieved. This approach, then, frequently provides low “recall” rates since applicable documents can remain unretrieved.

In an effort to improve the precision and recall of text management systems, a numerical paradigm commonly referred to as the “vector space model”, has been proposed. Early work by Salton [1,2,3], and later work by Salton and Buckley[4] described a model for text where each document was represented by a vector. The vector space consisted of a number of coordinates that was equal to the number of unique terms in the system (the “vocabulary”). The direction of the document vector was determined by counting the number of terms in the document, then applying an appropriate term weighting and normalization [4,5] to these counts. This approach produces what we will refer to as a “term orthogonal” space. That is, each term in the vocabulary is a coordinate and all coordinates are, by definition, orthogonal.

This approach results in several unavoidable consequences. First, is the high dimensionality of the resulting vector space. Second, there is no provision for encoding similarity of meaning at the term level. Given the availability and capacity of low cost, high performance computing resources, the first consequence (high dimensional space) has little practical impact and can be ignored. The second consequence, term orthogonality and inability to represent similarity of meaning, however, remains.

Extensions to the term orthogonal model have been proposed and demonstrated in the form of Latent Semantic Indexing (LSI) [6]. The basic LSI model extracts information in vector form by identifying the dependencies between terms and documents. This approach is based on constructing and processing a term-by-document matrix. In LSI, the resulting dimension of the vector model is determined via singular-value decomposition of the term-by-document matrix and, in general, will be significantly smaller than either the term or document count (typically 100 - 300). As a consequence, the vectors that represent words cannot be orthogonal. As such, the LSI approach encodes a form of similarity of usage, but at a low resolution, term-by-document level. Efforts by Deerwester, Furnas, Dumais, et al, [6,7,8] demonstrated the viability of the non-term orthogonal approach and provided the motivation for the early MatchPlus concept as proposed by Gallant[9]. Fair Isaac's approach used in MatchPlus is, in a sense, a close relative of LSI in that:

1. The vector space is not term orthogonal.
2. Relationships are computed (learned) from a training corpus.
3. The resulting dimensionality of the vector space is much smaller than either the term or document count in the corpus.

However, the *MatchPlus* approach has several key differences:

4. Relationships are determined at a finer resolution than the document level. That is, proximity of occurrences is taken into account in the MatchPlus approach. This provides

sensitivity to similarity of usage at the word level. Word sense identification and disambiguation is a direct consequence of this sensitivity [10].

5. The MatchPlus learning algorithm is based on self organization and employs an adaptive neural network learning law. This is in contrast to LSI's matrix manipulation approach which is a "block update" approach.
6. No orthogonal basis sets are derived in the MatchPlus technique. The learning technique chooses a coordinate space that is convenient for the algorithm and is unintelligible to the human.

The key innovations behind the *MatchPlus* approach are motivated by the desire to exploit neural network learning techniques to discover similarity of usage at the word level, in a language-independent manner, without the need for external dictionaries, thesauri or semantic networks. Additionally, since many domains use terms in a manner that is specific to that domain, we seek to have the *MatchPlus* algorithm determine these usages from the training corpus directly. Additionally, we seek to use the resulting word vectors to provide retrieval, routing, document clustering and summarization.

The Context Vector Concept

The key technical feature of MatchPlus is the representation of terms (stems), documents, and queries by context vectors. A context vector is a high dimensional vector consisting of real-valued numbers or components. All operations in MatchPlus are based on the geometry of these high dimensional spaces [11]. Specifically, closeness in the space is equivalent to closeness in subject content. The MatchPlus learning algorithm is designed to adjust word vectors such that terms that are used in a similar context will have vectors that point in similar directions.

Additionally, context vectors for documents with similar subject content will have vectors that point in similar directions. At the summary level, the MatchPlus system translates free text into a mathematical representation in a meaningful way. The resulting representation can be used for document retrieval, routing, document clustering and other text processing tasks. Since both terms and documents are represented in the same frame of reference, this allows several unique operations. Table 1 below describes the dualities of the context vector approach.

TABLE 1. CONTEXT VECTOR DUALITY

From --> To:	Words	Document
Words	Case 1: Examination of learned relationships	Case 3: Document summarization
Documents	Case 2: Document retrieval	Case 4: Query-by-example

Table 1 can be interpreted as follows:

Case 1. Given a word (or set of words), determine which word vectors are close in the context vector space. This action “queries” the learned relationships at the word level. More on this in Section 4.

Case 2. Given a word (or words), determine which document vectors are close to the word vectors. In this case the “words” are actually a user query and this operation is a document retrieval operation.

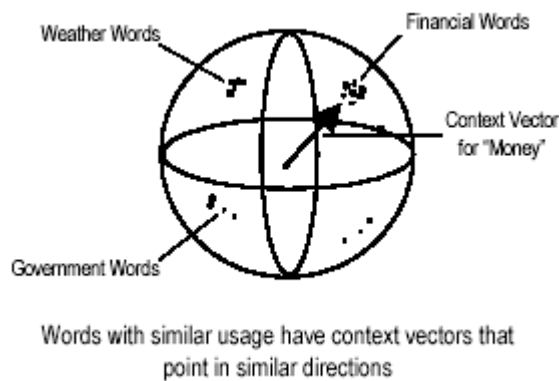
Case 3. Given a document, determine which word vectors are close to the document vector. This operation results in a summarization, or gisting, of the document.

Case 4. Given a document, determine which documents are close to the given document vector. This operation is actually “query by example” or document feedback.

Word Context Vectors

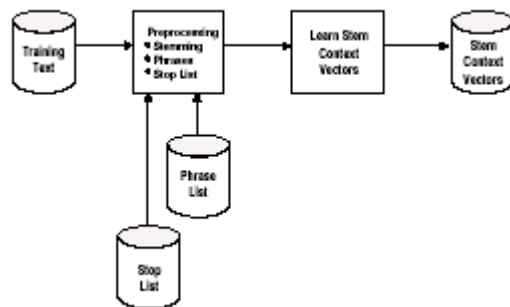
An approach has been developed to generate context vectors for words based on the context vectors of words that are “close” in context. This process of context vector generation based on the context vectors of words in similar context is called “bootstrapping”. Word context vectors which are initialized to random numbers with a gaussian random number generator to approximate mutual orthogonality are modified during the bootstrapping phase. The probabilistic influence of “close” words is determined over all instances of the word in the document corpus. In this way the system learns the associations between words by the context of their usage. In addition to being able to deduce the meaning of new words, the bootstrapping approach can be used to dynamically adapt to changing language usage patterns. The key result of this approach to word representation and machine learning is *terms that are used in a similar context will have vectors that point in similar directions*. A graphical example of this concept is shown in Figure 1.

FIGURE 1. MATCHPLUS CONTEXT VECTOR CONCEPT



The *MatchPlus* learning algorithm is based on the concept of “constrained self organization”. In this approach, constraints are applied to the learning algorithm to insure that the resulting vector space will possess the desired quality of “similar usage/similar direction” without allowing similarly used word vectors to get “too close”. A block diagram of the *MatchPlus* learning procedure is shown in Figure 2.

FIGURE 2. MATCHPLUS STEM VECTOR TRAINING



In practice, not all words from the document are used in the computation of the document context vector. In the current system, frequently used words such as “a, the, of”, etc. are removed prior to the determination of the context vector. These words are referred to as “stop words”. The *MatchPlus* system also has provisions for handling “word groups” or phrases. A word group is a sequence of words that should be treated as a single entity, such as “New York” or “gallium arsenide”. The system uses word groups determined from the training corpus to perform the correct associations. In addition to filtering the stop words and word group association, *MatchPlus* derives the root word, or “stem”, of each word prior to subsequent operations. This operation, referred to as “stemming”, eliminates plurality, tense, and suffixes, thus making learning easier. In Figure 2, note that *MatchPlus* does not use any external dictionaries, thesauri or knowledge bases to determine word vector relationships. These relationships are learned automatically using only the text examples provided during learning. The result of the learning procedure is a vocabulary of stem context vectors that can be used for a variety of applications.

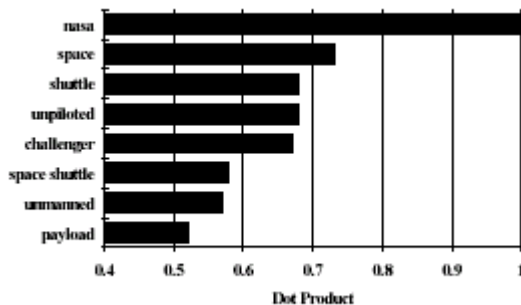
Assessment of Learned Relationships

To assess the nature of learned relationships, it is possible to determine which words are close to a selected word. This operation, called a *stem tree*, is performed as follows:

7. The user selects a “root” word for the tree and the trained context vector for that word is determined by a table lookup in the context vector vocabulary.
8. MatchPlus computes the distance of every other word vector in the vocabulary to the selected word.
9. The resulting closeness scores (normalized dot products) are sorted by magnitude (larger means closer in usage).

An example of a stem tree for the term *NASA* is shown in Figure 3 as learned from the Associated Press, 1989.

FIGURE 3. LEARNED RELATIONSHIPS FOR *NASA*.



Notice that in Figure 3, the length of the horizontal bar corresponds to the strength of the learned relationship (dot product). Also note that the words are ordered from strongest at the top to weakest at the bottom. NASA is related to itself with a strength of 1.0. Note that space shuttle is very close in this concept space to NASA. This is because in terms of context of usage in the Associated Press, they are very similar. Additionally, it can be seen from inspecting Figure 3 that Challenger and unmanned are related to NASA. These relationships were learned using only the training corpus. No external knowledge was used in the formations of these relationships.

Another example that demonstrates the ability of the MatchPlus learning algorithm to identify relationships is shown in Figure 4. Figure 4 is a stem tree for the term Hezbollah as learned from a subset of that in Figure 3, the length of the horizontal bar corresponds to the strength of the learned relationship (dot product). Also note that the words are ordered from strongest at the top to weakest at the bottom. NASA is related to itself with a strength of 1.0. Note that space shuttle is very close in this concept space to NASA. This is because in terms of context of usage in the Associated Press, they are very similar. Additionally, it can be seen from inspecting Figure 3 that Challenger and unmanned are related to NASA. These relationships were learned using only the training corpus. No external knowledge was used in the formations of these relationships.

Another example that demonstrates the ability of the MatchPlus learning algorithm to identify relationships is shown in Figure 4. Figure 4 is a stem tree for the term Hezbollah as learned

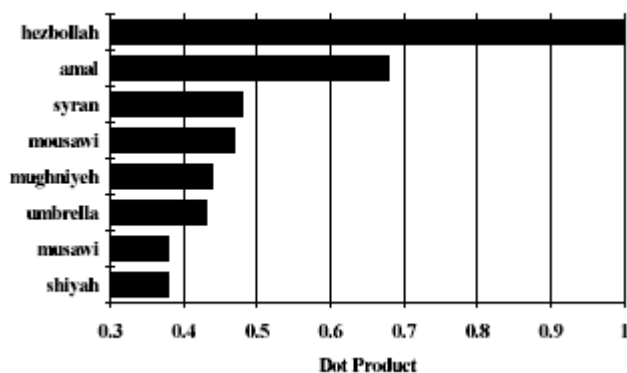
from a subset of the TIPSTER/TREC corpus (320 Megabytes of the Associated Press news wire, 1990). The context of more “obscure” references are shown in the text blocks that follow:

“ *Mousawi* heads the pro-Syrian Islamic *Amal* wing of the Hezbollah”

“...brother-in-law of Imad *Mughniyeh*, Hezbollah's security chief.”

“...*Musawi* heads the shiite Islamic Amal faction based in the...”

FIGURE 4. LEARNED RELATIONSHIPS FOR *HEZBOLLAH* WITH TERM CONTEXT FROM TRAINING TEXT.



This figure demonstrates the ability of the *MatchPlus* learning algorithm to identify associations from the text. In Figure 4 it can be seen that *MatchPlus* has not only learned the relationship of *Hezbollah* to *Amal* (the terrorist group), but has also established that *Mousawi* (Wall Street Journal spelling) is the head of the *Amal* wing of *Hezbollah* (also note the AP spelling “*Musawi*”), and that Imad *Mughniyeh* is also associated with the organization. This ability to discover relationships automatically from free text examples can be invaluable to the analysis of patents, news stories and military messages. Additionally, the stem tree approach is a novel technique for querying a free text database. This technique for exploring learned relationships can be used for linkage analysis (i.e automated hypertext links).

Generation of Document Context Vectors

Groups of words (text passages and queries) and documents can also be represented using the same context vector approach. Document context vectors are derived as the weighted sum of the context vectors associated with words in the document. Document context vectors are normalized to prevent long documents from being favored over short documents. This document context vector generation process is diagrammatically shown in Figure 5.

The system generation data flow diagram for MatchPlus is shown in Figure 6. Note that in Figure 6, the same preprocessing involving stemming and phrase identification is performed during document context vector generation as used during the stem vector learning process (refer to figure 2). Also note that the vocabulary of trained stem vectors that were learned in the bootstrapping step are applied to the user's text corpus to generate the document context vectors.

FIGURE 5. DOCUMENT CONTEXT VECTOR GENERATION FLOW DIAGRAM.

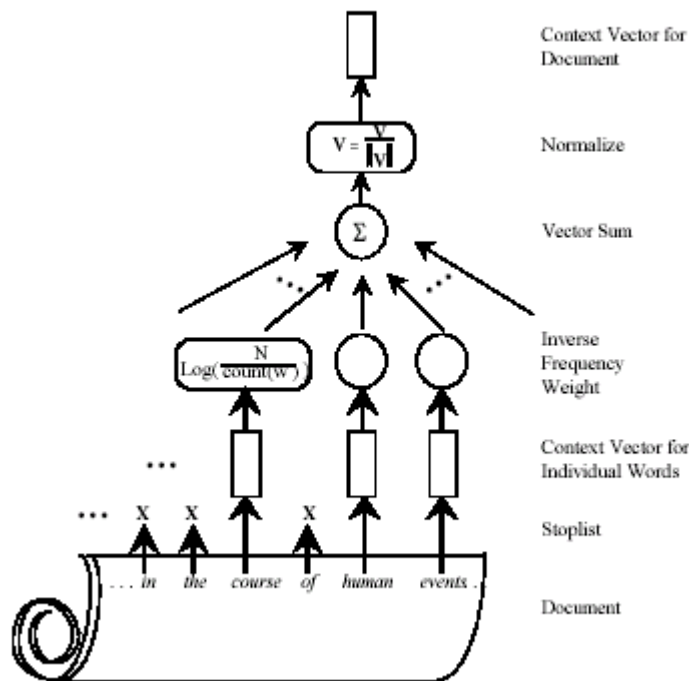
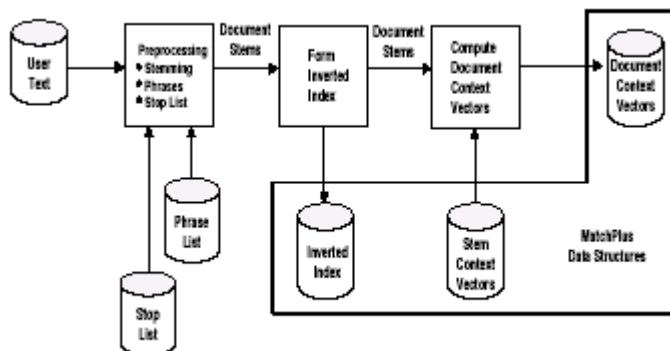


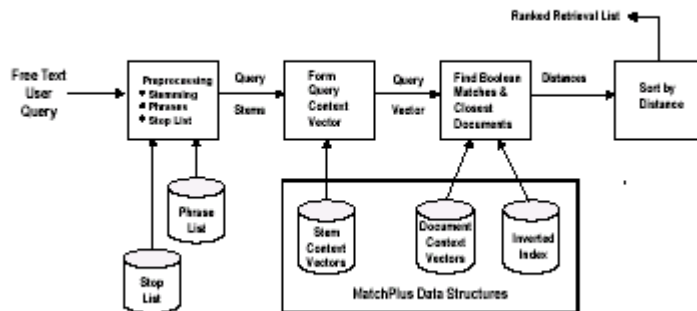
FIGURE 6. MATCHPLUS SYSTEM GENERATION.



Document Retrieval

Document retrieval is performed by converting the user free text query to a context vector. Then, the *MatchPlus* system determines relevant documents by finding those document context vectors that are closest to the query vector. The flow diagram for this operation is shown in Figure 7.

FIGURE 7. MATCHPLUS RETRIEVAL DATA FLOW



In Figure 7, note that the same preprocessing operations of stemming, and phrase identification are performed. User free text queries are transformed into a context vector. Then, using the inverted index and the database of document context vectors, the relevance of each document is determined by dot product. The resulting list is sorted by relevance to produce the final ranked list of relevant documents that are presented to the user.

Ad Hoc Retrieval and Routing Performance

US Government testing of the MatchPlus system has yielded results for both precision and recall that demonstrate the viability of the context vector approach. These results were established at the second Text Retrieval Evaluation Conference (TREC-2). TREC-2, sponsored by the U.S. Government (NIST) in September 1993, was conducted to evaluate various text retrieval technologies and consisted of 16 university and commercial participants. This test was conducted on a Government-issued corpus of approximately 2 Gigabytes. This 766,000 document corpus consisted of heterogeneous text from the Wall Street Journal, Associated Press Newswire, Federal Register, Department of Energy Technical Abstracts, and Ziff-Davis computer publications.

Retrieval performance was assessed using 50 Government-supplied natural language queries. Criteria for evaluation consisted of the total relevant documents in the top 1000 retrievals for each of the 50 queries, average precision over 50 queries at 5 documents in the retrieval list, precision at 15 documents and the precision at 100 documents in the list. To perform the scoring, NIST used a team of retired intelligence analysts to determine the relevance of retrieved documents. In the course of the evaluation, over 100,000 documents were read and judged by the NIST scoring team. This level of effort has resulted in a firm foundation for assessing the quality of various text retrieval techniques in terms of the objectivity of the test, the size of the corpus and the statistical significance of the results. Retrieval and routing performance of the MatchPlus system, as demonstrated on the TREC-2 collection and a description of the runs submitted, are as follows:

A. Training Stem Vectors

A subset of the entire TREC/TIPSTER corpus was chosen for derivation of a training corpus. This corpus consisted of approximately 320 Megabytes of sampled text from the entire 2 gigabytes of text. This training corpus resulted in approximately 120,000 stem vectors being trained. The stem count is in contrast to the 620,000 stems that comprise all of the TREC corpus. The *MatchPlus* neural network learning algorithm was used to determine a context vector representation for each of the 120,000 stems based on their context in the training corpus. When the learning operation was complete, the resulting stem vectors were transplanted (i.e. the trained vectors for the "subset" stems were copied into the stem file for the full TREC corpus) and document context vectors were computed. The resulting stem and document context vectors were used for the TREC-2 retrieval and routing experiments. It should be noted that nearly 82% of the stems that occurred in the full corpus were not present in the training corpus. As such, they were assigned a *zero* vector. Stated differently, only 18% of the stem vocabulary was used as the basis for the generation of document context vectors.

B. Ad Hoc Retrieval Experiments

Two runs were submitted as part of the ad hoc retrieval experiments. The first run was a totally automated run, the second involved relevance feedback.

The automated run used the entire government supplied natural language query converted to a context vector. A broad boolean filter (i.e. required few query terms to be present in a document) was used. It should be noted the supplied queries had an average of 30 terms which caused tens to hundreds of thousands of documents to contain "boolean hits", supporting the runs were submitted as part of the ad hoc retrieval experiments. The first run was a totally automated run, the second involved relevance feedback.

The automated run used the entire government supplied natural language query converted to a context vector. A broad Boolean filter (i.e. required few query terms to be present in a document) was used. It should be noted the supplied queries had an average of 30 terms which caused tens to hundreds of thousands of documents to contain "Boolean hits", supporting the earlier claim that Boolean systems often times return far too many documents to effectively review. The filter had the effect of placing those documents that contained at least 4 of the terms above all other documents in the ranking. For all documents, context vector dot products were computed to determine document relevance and determine those documents in the submission list. This entire set of processing actions was totally automated and involved no human interaction. For the relevance feedback experiment, a totally automated run was performed as described above. Then the top 20 documents for each of the 50 queries was read and judged relevant/not relevant. The context vector for all relevant documents for a query were summed, normalized and added to the totally automated query vector. The resulting vector was used as the basis for a context vector-only (no boolean match) query and all corpus documents were relevance ranked by dot product. Results for both ad hoc runs are shown in the table below.

TABLE 2. MATCHPLUS TREC-2 ADHOC SCORES

Performance Metric	Ad hoc: Fully Automated	Ad hoc: Relevance Feedback
Relevant Ret.	6944	6911
Prec. at 10	.5680	.5680
Prec. at 100	.4520	.4784
Avg. Prec.	.2787	.2882

C. Routing Experiments

Routing was performed on a subset of the TREC corpus in which relevance judgements for documents were available for the 50 routing queries (For TREC the definition of routing is identical to an adhoc retrieval with the added advantage of having access to relevant and non-relevant documents to adjust or create your query accordingly).. Two routing approaches were used. Both approaches were based on neural network learning techniques used to determine weights based on relevance-judged (relevant and non-relevant) documents. They are described below. The stem weight learning approach uses a perceptron learning technique to compute weights for terms in a query, with one weight associated with each term in the query. As such, the network has one input for each term. Every judged document in the corpus provides an example to the network training algorithm. When the learning was complete, the resulting weights were then used as term weights for normal context vector query processing.

The full context vector learning approach is similar to the stem weight technique. Here, we attempted to learn an entire query context vector rather than just weights for stems. In this approach, document vectors are used as the inputs to the network and the relevance of the associated document is used to determine the weight adjustment (i.e. vector element). Weight adjustment is performed for all relevance-judged documents. The resulting weights are used directly as a routing query vector.

The two routing runs submitted to TREC further utilized the idea of "data fusion" or "mixing" [13]. Multiple retrieval runs are combined to produce a single retrieval list.

The first submission, called "Best Candidate", was composed of results from four sources: the two neural network techniques described above, the fully automated query (the first ad hoc submission) and the relevance feedback query (the second ad hoc submission). For each of the

50 queries, the best scoring source was determined by precision and recall numbers on a separate test corpus. . The best candidate's top 1000 documents were then used as the submission for that query. This was repeated for each query.

The second routing submission, called "Mix and Match", was based on a weighted "mix and match" approach. In this run, a submission for each query was formed by combining the results from each of the four methods on a document-by-document basis again using precision and recall numbers from a test corpus. Then, each document from each source was given a modified relevance score proportional to the quality estimate of that source and inversely proportional to document ranking. The final results list was formed by combining results document-by-document from each list. Results for both routing runs are shown in table 3.

TABLE 3. *MATCHPLUS* TREC-2 ROUTING SCORES

Performance Metric	Routing: Best Candidate	Routing: Mix and Match
Relevant Ret.	5833	6407
Prec. at 10	.6020	.6100
Prec. at 100	.4228	.4090
Avg. Prec.	.2810	.2927

We were very pleased with the TREC-2 results, given that the MatchPlus system is in its second year of existence. These results demonstrate the viability of the context vector approach. We expect the quality of results to improve as research on context vectors continues.

Document Clustering and Cluster Trees

A key capability of the MatchPlus system is the ability for context vector storage and retrieval using cluster trees. This data structure and associated algorithm allows rapid access to individual documents in large free text data bases.

The cluster tree is organized by similarity of meaning and is the result of a clustering operation on document context vectors. Recall that in the MatchPlus context vector space, documents with similar information content will possess context vectors that point in similar directions. Using this fact, it is possible to apply a clustering algorithm to find groups of documents with similar themes. The result of the clustering is a set of context vectors that point to the center of each document cluster. These clusters are, in essence, a subject index for the corpus. An additional benefit of the context vector approach is that once the vectors for the cluster centers have been computed, clusters can be explained automatically. This operation consists of determining the words whose vectors are closest to each cluster center. The resulting lists provide a summarization of the information content in each cluster. This capability is unique to MatchPlus and is possible because in the MatchPlus system, both word vectors and document vectors co-exist in the same information space. Conceptually, finding documents that are close to words is a retrieval. Finding words that are close to documents is summarization (see Table 1.). It is the combination of Fair Isaac's proprietary context vector approach and the similarity of meaning indexing capabilities provided by the cluster tree that can serve as the basis for construction of a document classification system.

Once context vectors for all documents in the system have been derived, a cluster tree can be generated to allow rapid access to these documents. The Fair Isaac cluster tree technique allows access to documents in sublinear time and allows a concise storage mechanism for the context vectors of the documents in the system (i.e. the time required for document retrieval increases much more slowly than the number of documents). Additionally, the cluster tree provides the vehicle for efficient nearest-neighbor searches of the document meaning space. This approach allows effective searches at modest time costs.

Document Classifier Determination and Automatic Classification

The cluster tree used by *MatchPlus* can provide a basis for a hierarchical classification system based on information content of the documents. This capability can be thought of as an automated, self organizing subject index. Specifically, each node in the tree can be assumed to be a classifier and can be assigned a unique identifier. All leaf nodes can be accessed by this classification and inherit the attributes of the parent node. This approach offers several advantages:

1. **Automation.** Once document context vectors have been determined, classification is determined by finding the "path of closest distance" through the tree. The path, then, represents the document classification. Statistical measurement techniques can be employed to determine when the current tree and associated classification scheme is obsolete.
2. **Meaning driven classification.** Nodes are determined using clustering techniques. As such, the classification scheme is "meaning driven" since the context vectors encode a representation of the meaning of the documents. No human intellectual effort is used in the determination of category classifiers.
3. **Self Explaining Clusters.** Clusters are "explained" by determining the word vectors that are closest to each cluster center. The closest words provide a summarization of the contents of each cluster.

Conclusions and Comments

The MatchPlus system and its associated neural network learning concepts are a relative newcomer to the world of information retrieval. The MatchPlus concept provides an approach to implementing retrieval by similarity of meaning and automatic classification of documents. This system solves a number of the shortcomings of Boolean search and offers significant increases in effectiveness of document searches over conventional Boolean approaches. The excellent TREC-2 performance scores demonstrate the commercial viability of the context vector approach as a standard retrieval engine.

Research sponsored by ARPA as part of the TIPSTER Phase I effort has enabled the discovery and development of many high value capabilities of the context vector approach. However, more research is necessary to fully exploit the concepts described above. Additional efforts are planned to refine and exploit the potential of the context vector approach. Specifically, explanation of differences between documents and document clusters in the context vector concept space is currently being conducted. Improvements to the constrained self organization learning technique are being researched as well. This work was sponsored in part by ARPA under contract number 91-F136300-

References

- [1] Salton, G. (ed.), The SMART Retrieval System - Experiments in Automatic Document Processing, Prentice-Hall, 1971.
- [2] Salton, G., Another Look at Automatic Text Retrieval Systems, Communications of the ACM, Vol. 20, 1986, pp. 648 - 656.
- [3] Salton, G., Automatic Text Processing, Addison-Wesley, 1989.
- [4] Salton, G. Buckley, C. Term Weighting Approaches in Automatic Text Retrieval, Information Processing and Management, Vol. 24, No. 5, 1988, pp 513-523.
- [5] Buckley, C., Allan, J., Salton, G., Automatic Routing and Ad-hoc Retrieval Using SMART: TREC-2, in Proceedings TREC-2 Conference, D. Harman, ed, Gaithersburg, MD. Aug. 1993.
- [6] Deerwester, S., Dumais, S.T., et al, Indexing by Latent Semantic Analysis, Journal of the Society for Information Science, 1990, Vol 41, No. 6, pp 391-407.
- [7] Dumais, S.T., LSI Meets TREC: A Status Report, in Proceedings TREC-1 Conference, D. Harman, ed, NIST Special Publication 500-207, Rockville, MD. Nov. 1992.
- [8] Dumais, S.T., Latent Semantic Indexing (LSI) and TREC-2, in Proceedings TREC-2 Conference, D. Harman, ed, Gaithersburg, MD. Aug. 1993.
- [9] Gallant, S.I., Context Vector Representation for Document Retrieval, AAAI-91 Natural language text Retrieval Workshop, Anaheim, CA. July 1991.
- [10] Gallant, S.I., A Practical Approach for Representing Context and Performing Word Sense Disambiguation Using Neural Networks, Neural Computation, Vol. 3, No. 3, 1991, pp 293-309.
- [11] Watson, G.S., "Statistics on Spheres", John Wiley and Sons, 1983.
- [12] Gallant, S.I., Caid, W.R., et al, "HNC's *MatchPlus* System", in Proceedings TREC-1 Conference, D. Harman, ed, NIST Special Publication 500-207, Rockville, MD. Nov. 1992.
- [13] Gallant, S.I., Caid, W.R., et al, "Feedback and Mixing Experiments with *MatchPlus*", in Proceedings TREC-2 Conference, D. Harman, ed, Gaithersburg, MD. Aug. 1993.

About Fair Isaac

Fair Isaac Corporation (NYSE:FIC) is the preeminent provider of creative analytics that unlock value for people, businesses and industries. The company's predictive modeling, decision analysis, intelligence management, decision management systems and consulting services power more than 25 billion mission-critical customer decisions a year. Founded in 1956, Fair Isaac helps thousands of companies in over 60 countries acquire customers more efficiently, increase customer value, reduce fraud and credit losses, lower operating expenses and enter new markets more profitably. Most leading banks and credit card issuers rely on Fair Isaac solutions, as do insurers, retailers, telecommunications providers, healthcare organizations and government agencies. Through the www.myfico.com Web site, consumers use the company's FICO[®] scores, the standard measure of credit risk, to manage their financial health. For more information, visit www.fairisaac.com.

Corporate Headquarters:

200 Smith Ranch Road
San Rafael, CA 94903-5551
1 800 999 2955 *from the US*
1 415 472 2211 *from anywhere*
info@fairisaac.com *email*

Offices Worldwide:

Brazil, Canada, France, Germany,
Japan, Singapore, Spain,
United Kingdom, United States

www.fairisaac.com



Fair Isaac is a trademarks of Fair Isaac Corporation, in the United States and/or in other countries. Other product and company names herein may be trademarks or registered trademarks of their respective owners.

Copyright © 2003 Fair Isaac Corporation. All rights reserved.