Information Retrieval Using the Reduced Row Echelon Form of a Term-Document Matrix

Ufuk Parali[1] , Metin Zontul[2], Duygu Celik Ertugrul[3]

[1] Department of Electrical and Electronics Engineering, Ardahan University, Turkey
[2] Department of Computer Engineering, Istanbul Arel University, Turkey
[3] Department of Computer Engineering, Eastern Mediterranean University, Turkey
ufukparali@ardahan.edu.tr, metinzontul@arel.edu.tr, duygu.celik@emu.edu.tr

Abstract

It is getting more difficult to retrieve relevant information regarding the user input query due to the large amount of information in the web. Unlike the conventional information retrieval (IR) algorithms, this study presents a new algorithm – reduced row echelon form IR method (rrefIR) – with higher average similarity precision to get more relevant and noise-free documents. For dimension reduction in the proposed algorithm, singular value decomposition (SVD) is applied on the reduced row echelon form – obtained by utilizing Gauss-Jordan method – of the covariance of term-document matrix (TDM). The rrefIR algorithm outperforms the LSI and COV algorithms with respect to Jaro-Winkler, Overlap, Tanimoto and Jaccard similarity measures in the means of average similarity precision. The physical reason for the better IR performance is the linear independent basis vectors set obtained by Gauss-Jordan operation. This basis set can be considered as the generating roots of the vector space spanned by TDM. Utilizing these vectors increases the latent semantic charateristics of the SVD phase of the proposed IR algorithm.

Keywords: Information retrieval, Gauss-Jordan, SVD, Similarity measures

1 Introduction

Information retrieval is defined as finding the materials (documents) of an unstructured nature (text) that satisfies an information requirement from within large collections when a user enters a query into the system [1]. In IR problems, a numeric score is calculated to show the matching level of each object in the database with user input query so that the top ranking objects can be shown to the user [2].

In large amounts of available information and a high rate of new information updated, a low signal-to-noise ratio and inefficient methods for comparing and processing different kinds of information are the nowadays problems of the web [3-4]. Thus, it is getting more difficult to retrieve relevant information regarding the user input query. Information retrieval consists of translating and matching a query against a set of information objects where the IR system responds to the query using a given algorithm and a similarity measure [5]. A semantic similarity measure is a function that can be used to assign a numeric value to the similarity between two classes of objects based on the meaning related to each of the objects [6-7].

Deerwester et al. [8] and Berry et al. [9] have proposed latent semantic indexing (LSI) that uses truncated singular value decomposition or principal component analysis to discover latent relationships between correlated words and documents.

Kobayashi et al. [10] proposed a novel information retrieval algorithm for massive databases based on vector space modeling and spectral analysis of the covariance matrix, for the document vectors, to reduce the scale of the problem. They indicated that their algorithm COV was more accurate than the previous algorithm LSI.

In another study, Rölleke et al. [11] presented a well-defined general matrix framework for modeling information retrieval. In their framework, documents and queries were expressed in terms of matrix spaces in which concepts were defined with respect to the semantic relationships considering parent–child matrices that represent the relationship between documents or terms.

According to the approaches by Furnas et al. [12] and Jones et al. [13] for automatic indexing and retrieval, the implicit higher-order structure in the association of terms with documents is modeled to improve estimates of term-document association, and therefore the detection of relevant documents on the basis of terms found in queries. The proposed model is based on a generalization of the factor-analytic model, called “two-mode factor analysis”, based on SVD, which can represent both terms and documents as vectors in a space of controllable dimensionality,

*Corresponding Author: Ufuk Parali; E-mail: ufukparali@ardahan.edu.tr
DOI: 10.3966/160792642019072004004

where the inner-products between points in the space gives their similarity.

One of the latest studies by Guan et al. [14] presents the Imprecise Spectrum Analysis (ISA) to accomplish fast dimension reduction for document classification by following the one-sided Jacobi method for computing SVD and simplifying its intensive orthogonality computation. This method uses a representative matrix composed of top-k column vectors that are derived from the original feature vector space and reduces the dimension of a feature vector by computing its product with this representative matrix.

Gao et al. [15] proposed to divide a large inhomogeneous dataset into several smaller ones with clustered structure, on which they applied the truncated SVD. Their experimental results showed that the clustered SVD strategies might enhance the retrieval accuracy and reduce the computing and storage costs.

In one of the recent studies, Jun et al. [16] have performed dimension reduction by combining SVD and principal-component analysis (PCA) to overcome the sparseness in document data clustering.

Tai et al. [17] proposed a method to improve retrieval performance of the vector space model in which high dimensional and sparse vectors were reduced by SVD and transformed into a low-dimensional vector space, namely the space representing the latent semantic meanings of words. They proved by experimental data that their model improved LSI model and provided an approach that makes it possible to preserve user-supplied relevance information for the long term in the system in order to utilize this information in the latter steps.

Efron [18] analyzed the statistical relationship between LSI and vector space model for IR. His analysis focused on each method's basis in the least-squares optimization. According to his study, retrieval was to be understood as a simplified classification problem while LSI was to be understood as a biased regression technique, where projection onto a low dimensional orthogonal subspace of the documents reduce model variance.

In another study, Thorleuchter and Poel [19] have used semantic classification by applying LSI together with a rank validation procedure to calculate conditional cross-impact probabilities.

There are some recent studies related to similarity measures and information retrieval based on TDM. Jimenez et al. [36] studied the soft cardinality over similarity measures to enhance their performance in terms of semantic similarity while comparing word vectors in natural language processing. They proved their models theoretically and empirically based on random data. Kocher and Savoy [37] used similarity measures to classify the documents based on author demographics such as gender and age. They combined K-nearest neighbors classifier with 24 different similarity measures on TDM matrix including the most 200 frequent words with weighted frequencies. Gysel, Rijke and Kanoulas [38] have proposed the Neural Vector Space Model (NVSM) for unsupervised query-document matching on bag-of-words based TDM data. Instead of applying dimensionality reduction to TDM, the model learns representations directly by gradient descent. They have indicated their model outperforms the classical latent vector space models.

As explained above, one of the main tools in IR algorithms is the SVD operation due to its latent semantic characteristic [39]. The conventional IR algorithms directly apply SVD on TDM. However, instead of direct usage of TDM, if the lineer-independent basis vectors set of TDM were utilized in SVD, it would provide the retrieval of more relevant information without increasing SVD dimension.

The purpose of this study is to develop an IR algorithm with relatively increased average similarity precision to get more relevant and noise-free documents. Unlike the conventional IR algorithms such as LSI and COV, instead of applying SVD operation directly onto the user input TDM, we first applied Gauss-Jordan operation on TDM. This provides the reduced row echelon form containing the lineer-independent basis vectors set of TDM. Then, we applied the SVD operation onto this reduced row echelon form of the TDM.

The rest of the paper is organized as follows: Section 2 reviews the mathematical background of the approach for the proposed algorithm containing three important well-known mathematical models: Singular Value Decomposition, Latent Semantic Indexing and Covariance Matrix Analysis. Section 3 describes the working mechanism of the proposed IR algorithm. Section 4 gives the experimental results and discussions. Conclusions and proposed future studies are given in Section 5.

2 Mathematical Background

This section presents the mathematical background that is needed for understanding the explained details of the proposed algorithm in the next section.

2.1 Singular Value Decomposition (SVD)

SVD is a factorization method of an n × m real matrix A as given in Equation 1:

$$A = U \Sigma V^T \quad \textbf{(1)}$$

where U is an m × m real unitary matrix, Σ is an m × n rectangular diagonal matrix with nonnegative real numbers on the diagonal, and V^T is an n × n real unitary matrix. The entries along the diagonal of Σ, denoted as λ_1, ..., λ_r, are the singular values of A. The m columns of U and the n columns of V are called the left-singular vectors and right-singular vectors of A, respectively [20].

The SVD and the Eigen decomposition are closely related as defined below:

- The left-singular vectors of A are eigenvectors of AA^T.
- The right-singular vectors of A are eigenvectors of A^TA.
- The non-zero-singular values of A are the square roots of the non-zero eigenvalues of both A^TA and AA^T.

SVD is very general to be applied to any m × n matrix but eigenvalue decomposition can only be applied to certain classes of square matrices. However, the two decompositions are related as follows [21]:

$$A^TA = V\,\Sigma^T U^T U\,\Sigma\,V^T = V\,(\Sigma^T\Sigma)\,V^T \quad \textbf{(2)}$$

$$AA^T = U\,\Sigma\,V^T V\,\Sigma^T U^T = U\,(\Sigma\,\Sigma^T)\,U^T \quad \textbf{(3)}$$

The right-hand sides of these relations describe the eigenvalue decompositions of the left-hand sides as shown below:

- The columns of V (right-singular vectors) are eigenvectors of A^TA,
- The columns of U (left-singular vectors) are eigenvectors of AA^T.

2.2 Latent Semantic Indexing (LSI)

Latent Semantic Indexing (LSI) is an indexing and retrieval method that uses SVD to identify patterns in the relationships between the terms and concepts in the form of an unstructured collection of text. The LSI is based on the principle that words used in the same contexts tend to have similar meanings. The LSI has an ability to extract the conceptual content of a body of text by establishing associations between those terms occurring in similar contexts [22].

The LSI begins by constructing a term-document matrix A to identify the occurrences of the m unique terms within a collection of n documents. In a term-document matrix, each term is represented by a row, and each document is represented by a column. Since this matrix is usually very large and very sparse, the SVD is used for dimension reduction.

The SVD process used by the LSI decomposes the matrix into three matrices:

- U: a term by dimension matrix,
- Σ: a singular value matrix, and
- V: a document by dimension matrix.
- The number of dimensions is min (t, d) where,
- t: number of terms and
- d: number of documents.

In the LSI system, the U, Σ and V matrices are truncated to k dimensions. The dimensionality reduction reduces noise in the term–document matrix resulting in a richer word relationship structure that reveals latent semantics present in the collection. After dimensionality reduction the term-document matrix can be approximated by $U_k(\Sigma_k\,\Sigma_k^T)\,U_k^T$ and $V_k(\Sigma_k^T\,\Sigma_k)\,V_k^T$ [23].

2.3 Covariance Matrix Analysis (COV)

Given an n × m term-document matrix A, with n row vectors $\{d_i^T \mid i=1, 2,\ldots,n\}$ representing unique terms, each having m dimensions representing documents, the covariance matrix for the set of term vectors is defined as

$$C = (1/n)\sum_{i=1}^{n} d_i\,d_i^T - \bar{d}\bar{d}^T \quad \textbf{(4)}$$

where d_i represents the i^{th} term vector and $\bar{d}$ is the component-wise average over the set of all term vectors [10] as follows, $\bar{d} = [\bar{a}_1\ \bar{a}_2\ \ldots\ \bar{a}_m]^T$, $d_i = [\bar{a}_{i,1}\ \bar{a}_{i,2}\ \ldots\ \bar{a}_{i,m}]^T$ and $\bar{a}_J = (1/n)\sum_{i=1}^{n} a_{i,J}$. Because the covariance matrix is symmetric, positive and semi-definite, it can be decomposed into the product $C = V\,\Sigma\,V^T$, where V is an orthogonal matrix which diagonalizes C**.** The diagonal entries of Σ are in monotone decreasing order going from top to bottom in a way that $diag(\Sigma) = (\lambda_1, \lambda_2, \ldots, \lambda_m)$, where $\lambda_i \geq \lambda_{i+1}$ for i=1,2,…,m. In order to reduce the dimension of the IR problem to k which is less than min(n, m), all term vectors and the query vector are projected into the subspace spanned by k eigenvectors $\{v_1, v_2, \ldots, v_k\}$ corresponding to the largest k eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ of the covariance matrix C [10].

3 Description of the Proposed Search Algorithm for Information Retrieval

3.1 Pseudocode of the Algorithm

The proposed method has the following inputs and outputs:

Inputs:

(i) Term-Document Matrix (TDM_{nxm}): It contains m number of n-dimensional document (information) vectors;

(ii) Query Vector (QV_{nx1}): It contains one n-dimensional user defined query vector;

(iii) Cosine Threshold Value: Single scalar value used as threshold for similarity detection between QV_{nx1} and the document vectors of TDM_{nxm} in the projected space.

Outputs:

(i) Selected document vectors due to their cosine value bigger than the Cosine Threshold Value.

The steps of the proposed algorithm are defined as follows:

1: **function** rrefIR (TDM_{nxm}, QV_{nx1}, Cosine Threshold Value)

2: Find the covariance matrix of TDM_{nxm}: cov_TDM

3*: Find the reduced row echelon form (rref) of covTDM: rref_cov_TDM

4: Apply SVD on rref_cov_TDM: rref_Basis
5: Project each of the n-dimensional document vectors in TDM_{nxm} on to k-dimensional (k<n) space by using the left-right singular vectors and the biggest first k singular values of rref_Basis: rref_Basis_k
6: Project the n-dimensional QV_{nx1} vector on to the k-dimensional space evaluated in Step 5: QV_k.
7: Perform the Cosine Similarity Measure-Calculate the cosine value between the QV_k (projected QV) and each column vector (projected document vector) of rref_Basis_k matrix. Check which projected document has cosine value bigger than Cosine Threshold Value.
8: Choose the corresponding document vectors in the original TDM_{nxm} for cosine values bigger than the Cosine Threshold Value in Step 7.
9: **end function**

3.2 Detailed Description of the Algorithm

We named the proposed information retrieval algorithm as rrefIR (arg1, arg2, arg3) where the function gets the term-document matrix (TDM_{nxm}), query vector (QV_{nx1}) and the Cosine Threshold Value as input arguments, arg1, arg2 and arg3, respectively. The input TDM does not always necessarily need to be square and symmetric. In that case, it is not possible to apply matrix diagonalisation directly [10, 24-25]. Thus, the rrefIR function first calculates the covariance matrix of the TDM, named as cov_TDM (Step 2), where the dimension of the covariance matrix is independent of the number of documents. This provides the usability of the covariance matrix of a set of thousands of document vectors in IR applications for relatively small number of terms [10]. Then, the function utilizes the Gauss-Jordan method for obtaining the reduced row echelon form of the covariance matrix of the input TDM, named as rref_cov_TDM (**Step 3***). This reduced form matrix can be considered as a map to discover the hidden relationships among the vector space that is spanned by the rows of cov_TDM which is row equivalent to rref_cov_TDM [26-27]. Since each vector in this vector space can also be built up by the rows of the rref_cov_TDM, this finite minimal generating set is a simplified spanning basis set with no unnecessary elements. Besides, all the elements in this set are linearly independent and required for spanning the vector space spanned by the rows of the cov_TDM [25-26, 28].

Although the third step is the main and the only difference of the proposed algorithm in this study comparing to the other common IR algorithms such as LSI [9] and COV [10] in the literature, rrefIR algorithm has a better retrieval performance in the means of average similarity precision due to the Step 3. Instead of applying the SVD directly on to the cov_TDM (or directly on to the TDM as in the LSI algorithm) to find a simpler approximation, we (i) first evaluate the rref_cov_TDM which is the reduced row echelon form of the cov_TDM, and then (ii) decompose this reduced form, utilizing SVD, into orthogonal matrices (rref_Basis) containing left and right singular vectors and diagonal matrix of singular values.

Reducing to row echelon form helps SVD to provide a better result for filtering out the noise from cov_TDM during the calculation of rref_Basis_k which is the reduced dimensional form of rref_Basis by factor k (k largest singular values). In other words, if we utilize rref_cov_TDM instead of cov_TDM (or TDM) in SVD for evaluating the approximate version with a lower rank matrix (named as rref_Basis at the 4th step of the algorithm), this will provide a better performance in eliminating the documents that use keywords in unwanted contexts [28]. Since deleting the terms with small signal-to-noise ratio (SNR) causes a loss of a small part of the total signal with the removal of a disproportionately large component of the total noise in the original database and this is why a truncated SVD can filter out some of the noise without losing significant information about the signal in the original data [27]. Thus, the terms of rref_Basis_k have relatively higher SNR ratio if we obtain rref_Basis_k from rref_cov_TDM instead of obtaining it from cov_TDM (or TDM).

The physical reason of this result is the linear independency of the vectors in rref_cov_TDM due to Gauss Jordan operation where at the end of this operation the basis set or the generating set of the vector space spanned by the cov_TDM (or TDM) is obtained. The vectors of this generating set can be considered as the generating root vectors of the vector space, and working with these generating vectors at the SVD phase of the information retrieval provides more noise-free, refine and relevant outcomes. In the next section, this argument are justified with a numerical experiment and the information retrieval outcomes of LSI, COV and rrefIR algorithms are compared with respect to some performance evaluation criterias. A noticeable performance difference in the means of average similarity precision is detected for rrefIR algorithm that exceeds the performances of LSI and COV.

4 Experimental Results and Discussions

Average similarity precision is one of the main measures used to summarize the retrieval performance of an IR algorithm [9]. It is defined as the proportion of relevant documents in the set returned to the user (Precision := (retrieved ∩ relevant) / retrieved) [11].

The starting point of our experiment is the TDM. The generation of the terms of the TDM and the selection of the user inputs among these terms are done randomly using a word generator. In this study, the

TDM is assumed to be a 40x40 matrix where the document column vectors are available as shown in Table 1 given below (In this work, we only used a relatively small and uniformly distributed randomly generated dataset just to compare the performance of the proposed algorithm with respect to the other IR algorithms. As a future study, we plan to perform the proposed rrefIR algorithm on a real bigdata environment). Here, the symbols from D1 to D40 represent the document column vectors in TDM. The document vectors contain 0s for missing terms and 1s for the included terms. The assignment of 0s and 1s for the terms in the QV and for the terms in each of the document with respect to absence or presence in the documents is also done randomly using uniformly distributed random number generator function, rand(), of MATLAB. Thus, since the QV and the document vectors of TDM are randomly generated, it is not possible for the results to be dataset-specific.

Table 1. The 40x40 term document matrix

TERM INPUT QV			TERM - DOCUMENT MATRIX																																						
			D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21	D22	D23	D24	D25	D26	D27	D28	D29	D30	D31	D32	D33	D34	D35	D36	D37	D38	D39	D40
island	-	**0**	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	0	0	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
drain	✓	**1**	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	1
accident	-	**0**	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
hairy	-	**0**	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0	1	0	0	1	0	1	0	1	1	0
cannibal	-	**0**	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
tense	-	**0**	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	1
ripe	-	**0**	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0
dynamite	-	**0**	0	0	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	0	0	1	0	0	1
inner	-	**0**	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	1	1	0	0	1	1	1	0	0	0	0	1	1	0	1	1	1
paper	-	**0**	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	1	1
plasma	-	**0**	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0
national	-	**0**	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0
greasy	-	**0**	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0
insect	-	**0**	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0	1	1
thin	-	**0**	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1
land	✓	**1**	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	0	0	0	1	1	0	0	0
parade	✓	**1**	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	1	0	1	0	0	0	0	0	0	1
chemical	✓	**1**	1	0	1	0	1	1	1	1	0	1	0	0	0	1	1	0	1	1	0	1	0	0	1	0	1	1	0	1	0	1	1	1	1	0	1	0	0	0	0	1
abstract	✓	**1**	0	0	1	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
number	✓	**1**	1	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	0	0
crater	-	**0**	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0
day	-	**0**	0	1	1	1	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	1	0	0	0	0
uptown	✓	**1**	0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	0	0	1	1	1	1	0	0
reverse	-	**0**	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	0	1	1
analyze	✓	**1**	0	1	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1
false	-	**0**	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0	0
torpedo	-	**0**	1	0	0	1	1	1	0	1	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	1	0
starfish	-	**0**	1	1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	1	1	0	0	0	0	0
spirit	-	**0**	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0	0	0
embrace	-	**0**	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	1
award	✓	**1**	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	1	0	1	1	0	1	1	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0	0
knuckles	-	**0**	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0
contrast	-	**0**	1	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	1	1	0	0	0	0	0
tongue	✓	**1**	0	0	1	1	1	0	1	0	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0
cotton	✓	**1**	0	0	0	1	0	1	1	1	0	1	0	1	1	0	1	0	1	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0	0	0	0	1	0	1	0	0
aviator	✓	**1**	0	1	0	1	1	0	1	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	1
chamber	-	**0**	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	1	0	1	1	1	0	1	1	1
ghetto	✓	**1**	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	0	1	1	0
orphan	-	**0**	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0	1	1	0	0	0	1	0	0	0	1	1	0	1	0
airspace	-	**0**	0	0	0	1	0	0	0	1	0	1	1	0	1	0	1	0	0	0	0	0	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	0	1	0	0	0

TDM, QV and the Cosine Threshold Value are utilized as the input arguments for each of the IR algorithm; LSI, COV and rrefIR as shown below in Figure 1. The Cosine Threshold Value (selected as 0.9 in this study) is determined with respect to the general assumptions in the literature [8-9]. Then each IR algorithm returns its own specific set containing retrieved documents exceeding the threshold value. The set of each IR algorithm containing retrieved documents are listed in Table 2 given below. Different cosine threshold value would give smaller or larger set of retrieved documents. However, the effect of the change in the threshold value is out of our scope in this study and we used only one value.

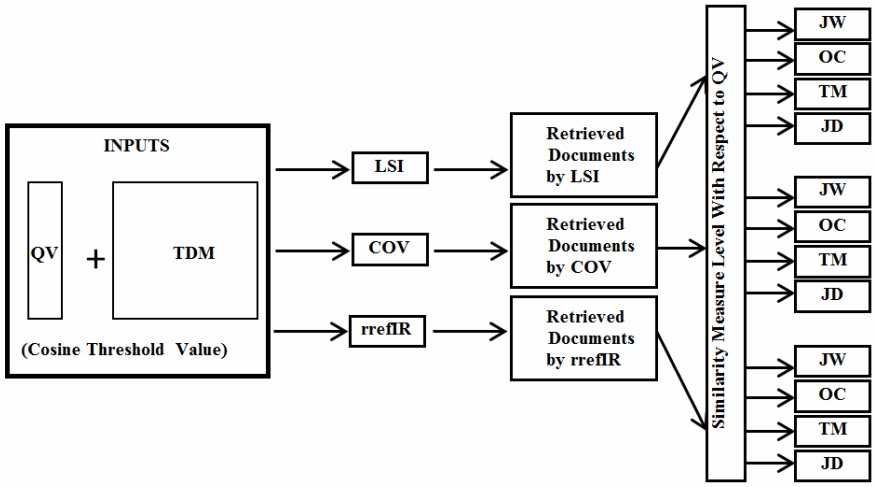

Figure 1. General combined flow diagram of information retrieval and similarity measure

Table 2. Document numbers retrieved by LSI, COV and rrefIR algorithms for the same cosine threshold value. Each number corresponds to the related document in TDM (i.e. number "3" means 3rd document (D3) in TDM matrix)

LSI	3	6	8	9	13	15	18	20	21	22	23	24	25	26	27	29	30	32	33	34	35	37	38	39	40
COV	2	3	4	5	9	11	12	13	15	16	17	18	31	32	36	37									
rrefIR	4	25	30	36	37																				
	Gray Scale 1							Gray Scale 2							Gray Scale 3										

To understand the retrieval performance of each algorithm, we need to calculate the similarity measure of each retrieved document with respect to the QV. A similarity measure is a function that evaluates the similarity between two objects, that refer to vectors in our experiment. We utilized the commonly used similarity functions explained briefly in Table 3 given below. We measure the retrieval performance applying four different functions (Jaro Winkler Similarity - JW, Overlap Coefficient Similarity - OC, Tanimoto Similarity - TM, Jaccard Similarity - JD) on each of the retrieved document vectors as shown in Figure 1. As an example of similarity calculation between a document vector and QV, we chose the 4th document vector – D4, as the sample retrieved document vector. The similarity results between D4 and QV for each of the similarity measure functions are shown in Table 4 below. In case of LSI, we take each of the retrieved 25 documents (the first row of Table 2) separately as the first vector and QV as the second vector. We calculate 25 similarity values for each of the retrieved 25 document vectors for each similarity measure (JW, OC, TM, JD). Then we find the average of these 25 similarity values calculated for each of the 4 measures. In case of COV and rrefIR, this time we apply the same procedure on each of the retrieved 16 documents (the second row of Table 2) and the retrieved 5 documents (the third row of Table 2), respectively.

Table 3. Similarity Measures

Jaro-Winkler [29]	$D_j + L.P.(1 - D_j)$	D_j: Jaro Distance [30], L: the prefix length, P: scaling factor
Overlap [31]	$\lvert A \cap B \rvert / \min(\lvert A \rvert, \lvert B \rvert)$	A: Query Vector, B: Document Vector $\lvert A \cap B \rvert$: Number of intersecting digit 1s
Tanimoto [32]	$\lvert A \cap B \rvert / (\lvert A \rvert + \lvert B \rvert - \lvert A \cap B \rvert)$	$\lvert A \cup B \rvert$: Number of digit 1s in A or B
Jaccard [33-34]	$\lvert A \cap B \rvert / \lvert A \cup B \rvert$	$\lvert A \rvert$: Number of digit 1s in A $\lvert B \rvert$: Number of digit 1s in B

Table 4. The comparison of 4th document vector and query vector (QV). Table A in appendix gives all LSI, COV and rrefIR performance results of each outcome for the same input term-document matrix

	4th document vector (D4) and user input Query Vector (QV) in TDM	Jaro Winkler	Overlap	Tanimoto	Jaccard
D4	0 0 0 1 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 **1** 1 0 **1 1 1** 1 0 0 1	0.5590	0.3076	0.1481	0.1538
QV	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 1 0 0 0 0 0 **1** 0 0 **1 1 1** 0 1 0 0				

In order to avoid computational cost, same SVD dimension value (factor k=2 as in [9]) is utilized in LSI, COV and rrefIR algortihms to retrieve the documents listed in Table 2. The document numbers given in gray scale 1 are the common documents retrieved only by both LSI and COV algorithms. The document 37 (D37), given in gray scale 2, is retrieved by all three of the algorithms. The document numbers given in gray scale 3 are retrieved by either both LSI and rrefIR or both COV and rrefIR. Table 2 shows that the number of relevant documents retrieved is maximum for the LSI and minimum for the rrefIR algorithm where the documents retrieved by the rrefIR are also retrieved by either LSI or COV (or retrieved by both of them). This indicates that there is no outlier result among the rrefIR outcomes. Furthermore, the striking point is that, as seen in Table 5 and Figure 2, the average similarity precision performance of the rrefIR outcomes with respect to the Jaro-Winkler (JW), Overlap coefficient (OC), Tanimoto (TM) and Jaccard (JD) similarity measures dominate the average similarity precision performances of the LSI and COV outcomes (only for Overlap coefficient similarity measure, the average similarity precision performance of the COV algorithm exceeds the performance of the rrefIR). The physical reason of the increment in the retrieval performance can be explained through the effect of the Gauss-Jordan operation embedded in rrefIR algorithm (Step 2 of the algorithm). As explained in Section 3, the linear-independent basis set provided by the Step 2 of rrefIR can be considered as the generating vectors of the vector space spanned by the TDM. Using these root vectors boosts the latent semantic characteristic of SVD phase by providing a better extraction of the interrelationships among the document vectors of the TDM.

Table 5. The average similarity precision performance results of LSI, COV and rrefIR algorithms

	JW	OC	TM	JD
LSI	0.6205	0.3844	0.2100	0.2158
COV	0.6316	0.4117	0.2126	0.2175
rrefIR	0.6591	0.4083	0.2211	0.2269

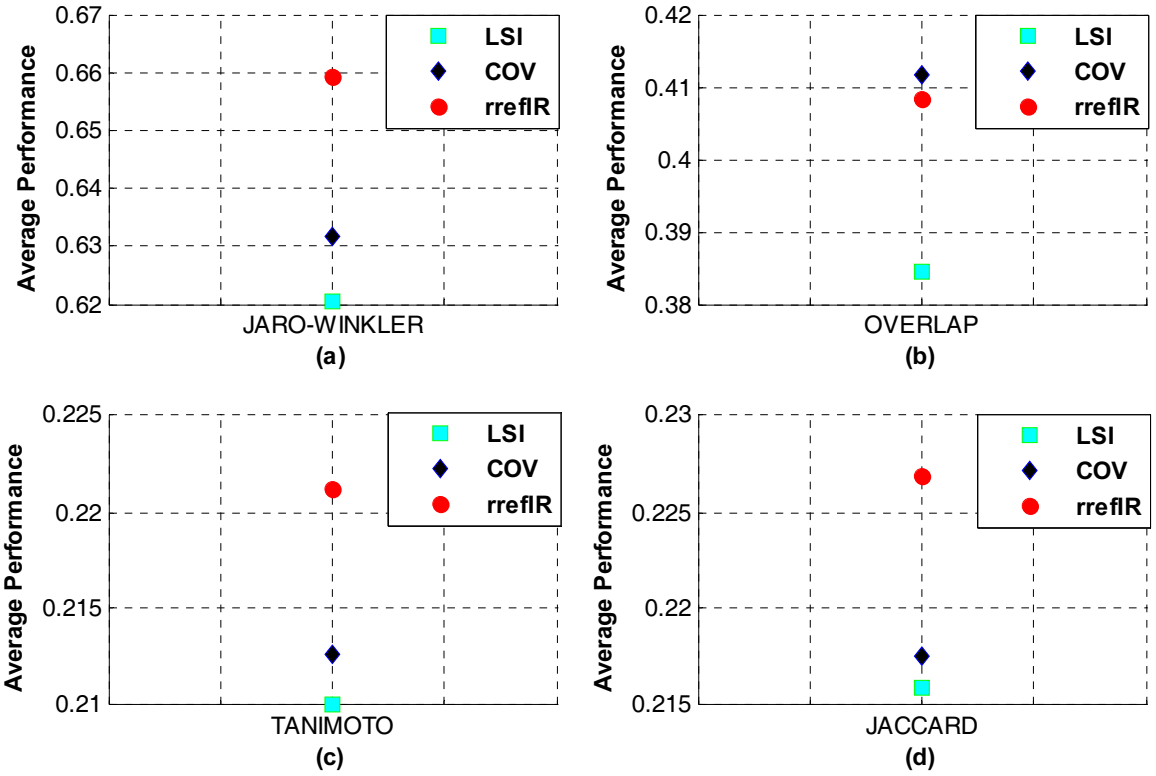

Figure 2. Separate graphical representation of average similarity precision performance results listed in Table 5

Choosing the value of k is an important issue in IR problems. While a reduction in the value of factor k can remove much of the noise, keeping the dimension lower than a reasonable level may cause to lose some vital information in the original input TDM matrix [9]. Thus, the objective in deciding the value of factor k for SVD based IR algorithms is to capture a major portion of the meaningful structure while eliminating the major portion of the noise. From Berry et al. [9], we know that the number of returned documents can significantly vary with the change in factor k where, as the dimension of SVD increases, the number of chosen documents decreases and documents with more relevancies regarding to the user input query are retrieved. Thus, keeping the dimension constant, it is novel to find a way to increase the noise cancellation while still capturing the same amount of the major portion of the meaningful structure. It is what the

rrefIR algorithm succeeds. Without changing the value of the factor k (dimension), rrefIR retrieves documents that contain information more related in the means of "meaning" with the user input query than the LSI and COV algorithms.

5 Conclusions

In this study, the conventional SVD based IR algorithm has been modified with the pre-applied covariance matrix and the Gauss-Jordan process for obtaining the reduced row echelon form (rref) of the term-document matrix. The proposed method is called rrefIR where the Gauss-Jordan method is applied on the covariance form of the term-document matrix before dimension reduction in SVD. The rrefIR algorithm outperforms the LSI and COV algorithms with respect to Jaro-Winkler, Overlap, Tanimoto and Jaccard similarity measures in the means of average similarity precision with the same SVD dimension. The linear-independency of basis vectors (that means; root vectors of the vector-space spanned by the input term-document matrix from knowledgebase) provided by Gauss-Jordan operation makes the rrefIR algorithm retrieve more noise-free and relevant documents than LSI and COV algorithms. Moreover, Gauss-Jordan helps the algorithm to get relatively more refined outputs without increasing SVD dimension (factor k). In addition, the rrefIR algorithm can be utilized practically in applications where SVD is involved. Thus, additional computational cost due to Gauss-Jordan algorithm in rrefIR is required. As a future study, we encourage the readership to reduce this cost using parallel Gauss-Jordan algorithm on conventional CPU parallel processing MPI platforms or newly developed GPU based CUDA platforms. And also, the application of rrefIR for bigdata analysis in cloud computing environment is another prominent and promising future study.

Acknowledgements

We would like to acknowledge Applied Mathematics conference (Amath–2014) where some part and results of this study has been presented [35].

References

[1] C. D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2009.

[2] B. Frakes, R. Baeza-Yates, *Information Retrieval Data Structures & Algorithms*, Prentice-Hall, 1992.

[3] W. Song, J. Z. Liang, X. L. Cao, S. C. Park, An Effective Query Recommendation Approach Using Semantic Strategies for Intelligent Information Retrieval, *Expert Systems with Applications*, Vol. 41, No. 2, pp. 366-372, February, 2014.

[4] X. Ning, H. Jin, H. Wu, RSS: A Framework Enabling Ranked Search on the Semantic Web, *Information Processing and Management*, Vol. 44, No. 2, pp. 893-909, March, 2008.

[5] S. Akmal, L. H. Shih, R. Batres, Ontology-based Similarity for Product Information-retrieval, *Computers in Industry*, Vol. 65, No. 1, pp. 91-107, January, 2014.

[6] T. A. Farrag, A. I. Saleh, H. A. Ali, Semantic Web Services Matchmaking: Semantic Distance-based Approach, *Computers & Electrical Engineering*, Vol. 39, No. 2, pp. 497-511, February, 2013.

[7] B. Hu, Y. Kalfoglou, H. Alani, D. Dupplaw, P. Lewis, N. Shadbolt, Semantic Metrics, *15th International Conference on Knowledge Engineering and Knowledge Management*, Prague, Czech Republic, 2006, pp. 166-181.

[8] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by Latent Semantic Analysis, *Journal of the American Society for Information Science*, Vol. 41, No. 6, pp. 391-407, September, 1990.

[9] M. W. Berry, S. T. Dumais, G. W. O'Brien, Using Linear Algebra for Intelligent Information Retrieval, *SIAM Review*, Vol. 37, No. 4, pp. 573-595, December, 1995.

[10] M. Kobayashi, M. Aono, H. Takeuchi, H. Samukawa, Matrix Computations for Information Retrieval and Major and Outlier Cluster Detection, *Journal of Computational and Applied Mathematics*, Vol. 149, No. 1, pp. 119-12, December, 2002.

[11] T. Rölleke, T. Tsikrika, G. Kazai, A General Matrix Framework for Modeling Information Retrieval, *Information Processing and Management*, Vol. 42, No. 1, pp. 4-30, January, 2006.

[12] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, K. E. Lochbaum, Information-retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure, *11th Annual Iternational ACM SIGIR Conference on Research and Development in Information Retrieval*, Grenoble, France, 1988, pp. 465

[13] W. P. Jones, G. W. Furnas, Pictures of Relevance: A Geometric Analysis of Similarity Measures, *Journal of the American Society for Information Science*, Vol. 38, No. 6, pp. 420-442, November, 1987.

[14] H. Guan, J. Zhou, B. Xiao, B. M. Guoi, T. Yang, Fast Dimension Reduction for Document Classification Based on Imprecise Spectrum Analysis, *Information Sciences*, Vol. 222, pp. 147-162, February, 2013.

[15] J. Gao, J. Zhang, Clustered SVD Strategies in Latent Semantic Indexing, *Information Processing and Management*, Vol.41, No. 5, pp. 1051-1063, September, 2005.

[16] S. Jun, S. S. Park, D. S. Jang, Document Clustering Method Using Dimension Reduction and Support Vector Clustering to Overcome Sparseness, *Expert Systems with Applications*, Vol. 41, No. 7, pp. 3204-3212, June, 2014.

[17] X. Tai, F. Ren, K. Kita, An Information-retrieval Model Based on Vector Space Method by Supervised Learning, *Information Processing and Management*, Vol. 38, No. 6, pp. 749-764, November, 2002.

[18] M. Efron, Query Expansion and Dimensionality Reduction:

Notions of Optimality in Rocchio Relevance Feedback and Latent Semantic Indexing, *Information Processing and Management*, Vol. 44, No. 1, pp. 163-180, January, 2008.

[19] D. Thorleuchter, D. Van den Poel, Quantitative Cross, Impact Analysis with Latent Semantic Indexing, *Expert Systems with Applications*, Vol. 41, No. 2, pp. 406-411, February, 2014.

[20] H. Gerlach, S. D. Blunt, The Factored-SVD Formulation and an Application Example, *Digital Signal Processing*, Vol. 17, No. 1, pp. 199-208, January, 2007.

[21] D. Kalman, A Singularly Valuable Decomposition: The SVD of a Matrix, *The College Mathematics Journal*, Vol. 27, No. 1, pp. 2-23, February, 1996.

[22] X. Chen, B. Gao, P. Wen, An Improved PageRank Algorithm Based on Latent Semantic Model, *International Conference on Information Engineering and Computer Science*, Wuhan, China, 2009, pp. 1-4.

[23] A. Kontostathis, W. M. Pottenger, A Mathematical View of Latent Semantic Indexing: Tracing Term Co-occurrences, *LeHigh Technical Report, LU-CSE-02-006*, June, 2002.

[24] M. Petrou, K. Petrou, *Image Processing-The Fundamentals*, Wiley, 2010.

[25] L. N. Trefethen, D. Bau, *Numerical Linear Algebra*, Siam, 1998.

[26] S. Andrilli, D. Hecker, *Elementary Linear Algebra*, Academic Press-Elsevier, 2010.

[27] C. D. Meyer, *Matrix Aalysis and Applied Linear Algebra*, Siam, 2000.

[28] S. J. Leon, *Linear Algebra with Applications*, Pearson, 2010.

[29] W. E. Winkler, The State of Record Linkage and Current Research Problems, *Statistical Research Division, US Census Bureau, RR99/04*, April, 1999.

[30] M. Jaro, Advances in Record Linking Methodology as Applied to Matching the 1985 Census of Tampa Florida, *Journal of the American Statistical Society*, Vol. 64, pp. 1183-1210, March, 1989.

[31] C. J. Rijsbergen, *Information Retrieval*, Butterworth - Heinemann, 1990.

[32] T. Tanimoto, *An Elementary Mathematical Theory of Classification and Prediction*, IBM Internal Report, November, 1958.

[33] P. Jaccard, Lois de Distribution Florale, *Bulletin de la Societé Vaudoise des Sciences Naturelles*, Vol. 38, pp. 67-130, 1902.

[34] S. S. Choi, S. H. Cha, C. C. Tappert, A Survey of Binary Similarity and Distance Measures, *Journal of Systemics, Cybernetics and Informatics*, Vol. 8, No. 1, pp. 43-48, February, 2010.

[35] U. Parali, M. Zontul, D. Celik, A New Information-retrieval Model Using Gauss-Jordan Method, *19th International Conference on Applied Mathematics*, Istanbul, Turkey, 2014, pp. 153-155.

[36] S. Jimenez, F. A. Gonzalez, A. Gelbukh, Mathematical Properties of Soft Cardinality: Enhancing Jaccard, Dice and Cosine Similarity Measures with Element-wise Distance, *Information Sciences*, Vol. 367-368, No. 1, pp. 373-389, November, 2016.

[37] M. Kocher, J. Savoy, Distance Measures in Author Profiling, *Information Processing & Management*, Vol. 53, No. 5, pp. 1103-1119, September, 2017

[38] C. E. Gysel, M. D. Rijke, E. Kanoulas, Neural Vector Spaces for Unsupervised Information-retrieval, *ACM Transactions on Information Systems*, Vol. 1, No. 1, pp. 1-25, January, 2018.

[39] J. Hu, L. Liu, C. Zhang, J. He, C. Hu, Hybrid Recommendation Algorithm Based on Latent Factor Model and PersonalRank, *Journal of Internet Technology*, Vol. 19, No. 6, pp. 919-926 May, 2018

Biographies

Ufuk Parali received his Ph.D. degree in Electrical Engineering from the University of Nebraska-Lincoln in 2011. He was a postdoctoral researcher at the Photonics group of Imperial College London where he worked on diode-pumped solid-state lasers. His research interests include DPSSL, EDFA, Ultrafast Laser-Matter Interaction, Big-Data and Linear Algebra.

Metin Zontul has Ph.D. on Numerical Methods, M.S. and B.S. on Computer Engineering, respectively. Currently, he is an Associate Professor at Istanbul Arel University. His research area includes: Artificial Neural Networks, Machine Learning, Big-Data, Data Mining and Sentiment Analysis. Dr. Zontul has 20 years of industrial experience as project manager.

Duygu Çelik Ertuğrul received her Ph.D. degree in Computer Engineering from Eastern Mediterranean University in 2010. Currently, she is an Associate Professor at Eastern Mediterranean University. Her research topics are Web Semantics, Composition & Discovery of Web Services, Semantic Agents, Rule-Based Expert Systems and e/m-Health.

Appendix

Table A. LSI, COV and rrefIR performance results for each retrieved document from the input TDM with respect to similarity measures JW, OC, TM and JD. The average results shown here are also listed in Table 5 and depicted in Figure 2.

LSI Model

Document No	JW	OC	TM	JD
3	0.7098	0.4545	0.2631	0.2631
6	0.4858	0.3000	0.1500	0.1578
8	0.6123	0.4000	0.2105	0.2105
9	0.5558	0.5000	0.1875	0.1875
13	0.3758	0.2857	0.1111	0.1111
15	0.6743	0.3846	0.2083	0.2500
18	0.7723	0.4615	0.2727	0.2727
20	0.8014	0.5454	0.3333	0.3333
21	0.6123	0.4000	0.2105	0.2105
22	0.4858	0.3000	0.1500	0.1500
23	0.6123	0.4000	0.2105	0.2352
24	0.6923	0.3846	0.2380	0.2500
25	0.4763	0.2727	0.1428	0.1578
26	0.7780	0.4615	0.2857	0.2857
27	0.5846	0.3076	0.1818	0.1818
29	0.6855	0.3846	0.2272	0.2272
30	0.8508	0.5384	0.3333	0.3333
32	0.8333	0.5384	0.2692	0.2800
33	0.6123	0.4000	0.2105	0.2105
34	0.5673	0.3076	0.1600	0.1600
35	0.5780	0.3076	0.1739	0.2000
37	0.5780	0.3076	0.1739	0.1818
38	0.7212	0.5000	0.2777	0.2777
39	0.1721	0.0833	0.0416	0.0416
40	0.6855	0.3846	0.2272	0.2272
AVG	**0.6205**	**0.3844**	**0.2100**	**0.2158**

COV Model

Document No	JW	OC	TM	JD
2	0.6257	0.4444	0,2222	0,2222
3	0.7098	0.4545	0.2631	0.2631
4	0.5590	0.3076	0.1481	0.1538
5	0.8558	0.5384	0.3500	0.3500
9	0.5558	0.5000	0.1875	0.1875
11	0.3415	0.2000	0.0952	0.0952
12	0.4975	0.3333	0.1578	0.1578
13	0.3758	0.2857	0.1111	0.1111
15	0.6743	0.3846	0.2083	0.2500
16	0.6423	0.5000	0.2352	0.2352
17	0.7003	0.4166	0.2500	0,2500
18	0.7723	0.4615	0.2727	0.2857
31	0.4858	0.3000	0.1500	0.1500
32	0.8333	0.5384	0.2692	0.2800
36	0.8995	0.6153	0.3076	0.3076
37	0.5780	0.3076	0.1739	0.1818
AVG	**0.6316**	**0.4117**	**0.2126**	**0.2175**

rrefIR Model

Document No	JW	OC	TM	JD
4	0.5590	0.3076	0.1481	0.1538
25	0.4763	0.2727	0.1428	0.1578
30	0.8508	0.5384	0.3333	0.3333
36	0.8318	0.6153	0.3076	0.3076
37	0.5780	0.3076	0.1739	0.1818
AVG	**0.6591**	**0.4083**	**0.2211**	**0.2269**