

## LETTER

# Probabilistic Information Retrieval Method Based on Differential Latent Semantic Index Space

Liang CHEN<sup>†</sup>, *Nonmember*, Naoyuki TOKUDA<sup>†</sup>, and Akira NAGAI<sup>†</sup>, *Members*

**SUMMARY** To improve the unstable performance of the traditional keyword-based search engine due to ambiguities of a natural language such as synonymy and /or polysemy, we have developed a new advanced DLSI (differential latent semantic index) space based probabilistic information retrieval system. The new method exploits a most likelihood posteriori function providing a measure of reliability in retrieving a document in the database having a closest match with another document of a query. Our simple experiment gives a supporting evidence for the validity of the theory, which is capable of capturing the intricate variability in word usage contributing to a more robust context contingent search engine.

**key words:** *Information Retrieval, Cross Language Information Retrieval, Differential Document Vector, Differential Latent Semantic Index Space*

## 1. Introduction

With the explosive usage of the Internet, an information retrieval system of higher performance level plays an increasingly more important role in processing the ever increasing amount of digital textual objects such as web pages and documents [7], [11], [12]. Many traditional query-based search engines are subjected to liability of an inherent ambiguity of the natural language resulting in an unstable retrieval performance because the choice of keywords as well as the method of summarization itself is ambiguous. As we all know, a considerable number of different words could be used to describe the same meaning (synonymy), and more often than not the same word could be associated with different meanings (polysemy) so that traditional keywords-based retrieval systems often miss important related materials to recall and/or recalling many unrelated documents. In image recognition applications [13], the dimension reducing capability is closely related to the features extraction capability as in PCA (principal component analysis). Sharing the same capability, the latent semantic structure with truncated singular vector decomposition (SVD) [1]–[3] is found to have a distinct effect in constructing a unified semantic space for retrieval, capturing not only most of the important underlying semantic structure in associating terms with documents, but also removing the noise or possible variability in word usage which consistently plagues the traditional keyword-based retrieval system.

The purpose of this paper is to verify the basic pos-

tulate of the features extraction function of the DLSI space by exploiting the differences of the document vectors on the primary reduced document space and its another secondary orthogonal space to the reduced document space.

It is worthwhile to demonstrate the difference between our new approach and the traditional LSI approach. To measure a similarity of the two documents, the traditional LSI method exploits the cosine measurement between the projections of a pair of normalized document vectors in the LSI space. The cosine measurement of projections of the two vectors actually has the same geometric meaning as the length of the projection of the differential document vector of the two documents in our differential latent semantic space. As pointed out by Hinrich Schütze and Craig Silverstein [6], an LSI is indeed a global dimensionality reduction approach and like all the global dimension reduction approaches the present method also encounters a difficulty in adapting to the unique particular characteristics of each document. In addition to the projection of the differential document vector, our new differential LSI-based method introduced can now make better use of the distance from the differential document vector to the differential LSI space by taking the “unique” characteristics of the difference between the pair of documents into account definitely capturing much richer information than the standard LSI based approach. After giving a brief introduction to the DLSI-space-based search algorithm in section 2, we give a simple example demonstrating how our method works effectively.

## 2. The DLSI Space Based IR Method

### 2.1 Differential term-document matrix

A term is defined as a word or a phrase that appears at least in two documents. The restriction of at least two documents here is added partially to remove some terms that are used very seldom, but most importantly it is capable of removing the words of somewhat erratic spelling errors. To reduce the possible number of candidates, the two documents can be increased to the three or more documents. It is a tradeoff problem between the computing time and computing resources and the number can be chosen accordingly depending on the applications sought. In the very special situa-

<sup>†</sup>The authors are with Computer Science Department, Utsunomiya University

tions when the database is very small like our present case of our case of section 3, the restriction may be removed by allowing a word or a phrase to be a term even if it appears only once in the database. We exclude the so-called stop words which are most frequently used in any topic, say "a", "the" in English. To effectively deal with morphological term variants of natural language, a stemming process is essential not only to reduce the size of indexing files but also to improve the efficiency of IR. A stemming algorithm needs be implemented before we set up the term index. Many stemming algorithms are available currently; But as noted by Lennon et al. [?, ], there seems relatively little difference as far as the final retrieval performance is concerned. We have used an Affix Removal Stemmer by Porter, named Porter algorithm [8]. Suppose we select and list the terms that appear in the documents as  $t_1, t_2, \dots, t_m$ .

Each document in collection will be assigned with a document vector as  $(a_1, a_2, \dots, a_m)$ , where  $a_i = f_i \times g_i$ , where  $f_i$  denotes a local weight for the number of times the term  $t_i$  appears in an expression of the document, and  $g_i$  denotes a global weight representing the importance of the term  $t_i$  applicable throughout the documents, which is a parameter to denote the importance of the term in representing the documents. Local weights could be either raw occurrence counts, boolean, or logarithm of occurrence count. Global counts may be given uniform weighting (uniform), domain specific, or entropy weighting. For example,

$$f_i = \log(1 + O_i)$$

and

$$g_i = 1 - \frac{1}{\log N} \sum_{j=1}^N p_{ij} \log(p_{ij}),$$

where  $p_{ij} = \frac{O_{ij}}{d_i}$ ,  $O_i$  is the number of times that the term  $t_i$  appears in the document,  $d_i$  is the total number of times that term  $i$  appears in the collection,  $O_{ij}$  the number of times that the term  $t_i$  appears in the document  $j$ ,  $N$  the number of documents in the collection. Notice that, we define  $p_{ij} \log(p_{ij})$  to be 0 if  $p_{ij} = 0$ . The document vector is normalized as  $(b_1, b_2, \dots, b_m)$  by the following formula:

$$b_i = a_i / \sqrt{\sum_{j=1}^m a_j^2}. \quad (1)$$

Because a summary of a document is obtained by any of various summarization techniques or a pre-assigned summary, we always associate the document with several other document vectors with a query to be regarded as a document.

We now define a differential document vector, which is used throughout the paper. An interior differential document vector is the differential document

vector defined as  $I = I_1 - I_2$ , where  $I_1$  and  $I_2$  are two different normalized document vectors of the same document. The different document vectors of the same documents may be taken from parts of documents, or may be produced by different schemes of summaries, or from the queries. Similarly an exterior differential document vector is defined as the Differential Document Vector  $I = I_1 - I_2$ , where  $I_1$  and  $I_2$  are two normalized document vectors of any two different documents. The interior differential term-document matrix and the exterior differential term-document matrix is defined as a matrix each of whose column is set to be an interior and an exterior differential document vector respectively.

## 2.2 Details of a General Model

Any differential term-document matrix, say, m-by-n matrix  $D$  of rank  $r \leq q = \min(m, n)$ , can be decomposed into a product of three matrices:  $D = USV^T$ , such that  $U$  and  $V$  are an m-by-q and q-by-n unitary matrices respectively, and the first  $r$  columns of  $U$  and  $V$  are the eigenvectors of  $DD^T$  and  $D^T D$  respectively.  $S = \text{diag}(\delta_1, \delta_2, \dots, \delta_q)$ , where  $\delta_i$  are non-negative square roots of eigen values of  $DD^T$ ,  $\delta_i > 0$  for  $i \leq r$  and  $\delta_i = 0$  for  $i > r$ .

By convention, the diagonal elements of  $S$  are sorted in a decreasing order of magnitude. To obtain a new reduced matrix  $S_k$ , we simply keep the k-by-k leftmost-upper corner matrix ( $k < r$ ) of  $S$  with the other terms deleted; we similarly obtain the two new matrices  $U_k$  and  $V_k$  by keeping the leftmost  $k$  columns of  $U$  and  $V$ . The product of  $U_k, S_k$  and  $V_k^T$  provides a matrix  $D_k$  which is approximately equal to  $D$ .

An appropriate value of  $k$  to be selected depends on the type of applications. Generally  $k \geq 100$  will be selected for  $1000 \leq n \leq 3000$ , and the corresponding  $k$  is normally smaller for the interior differential term-document matrix than that for the exterior differential term-document matrix.

Each of the differential document vector  $q$  could find a projection on the  $k$  dimensional fact space also called a differential latent semantic index space, as spanned by the  $k$  columns of  $U_k$ . The projection can easily be obtained by  $U_k^T q$ .

Note that, the mean  $\bar{x}$  of the exterior-(interior-)differential document vectors are approximately 0. Thus,  $\Sigma = \frac{1}{n} DD^T$ , where  $\Sigma$  is the covariance of the distribution computed from the training set. Assuming that the differential document vectors formed follow a high-dimensional Gaussian distribution, the likelihood of any differential document vector  $x$  will be given by

$$P(x|D) = \frac{\exp[-\frac{1}{2}d(x)]}{(2\pi)^{n/2} |\Sigma|^{1/2}},$$

where  $d(x) = x^T \Sigma^{-1} x$ . Since  $\delta_i^2$  are eigenvalues of  $DD^T$ , we have  $S^2 = U^T DD^T U$ , and thus

$$d(x) = nx^T(DD^T)^{-1}x = ny^T S^{-2}y,$$

where  $y = U^T x = (y_1, y_2, \dots, y_n)^T$ .

Because  $S$  is a diagonal matrix,  $d(x) = n \sum_{i=1}^r y_i^2 / \delta_i^2$ .

It is convenient to estimate the quantity by

$$\hat{d}(x) = n \left( \sum_{i=1}^k y_i^2 / \delta_i^2 + \frac{1}{\rho} \sum_{i=k+1}^r y_i^2 \right).$$

where  $\rho = \frac{1}{r-k} \sum_{i=k+1}^r \delta_i^2$ . In practice,  $\delta_i$  ( $i > k$ ) could be estimated by fitting a function (say,  $1/i$ ) to the available  $\delta_i$  ( $i \leq k$ ), or we could let  $\rho = \delta_{k+1}^2 / 2$  since we only need to compare the relative probability.

Because the columns of  $U$  are orthonormal vectors,  $\sum_{i=k+1}^r y_i^2$  could be estimated by  $\|x\|^2 - \sum_{i=1}^k y_i^2$ . Thus, the likelihood function  $P(x|D)$  could be estimated by

$$\hat{P}(x|D) =$$

$$\frac{n^{1/2} \exp\left(-\frac{n}{2} \sum_{i=1}^k \frac{y_i^2}{\delta_i^2}\right) \cdot \exp\left(-\frac{n\epsilon^2(x)}{2\rho}\right)}{(2\pi)^{n/2} \prod_{i=1}^k \delta_i \cdot \rho^{(r-k)/2}}, \quad (2)$$

where  $y = U_k^T x$ ,  $\epsilon^2(x) = \|x\|^2 - \sum_{i=1}^k y_i^2$ ,  $\rho = \frac{1}{r-k} \sum_{i=k+1}^r \delta_i^2$ ,  $r$  is the rank of matrix  $D$ . In practical cases,  $\rho$  may be chosen as  $\delta_{k+1}^2 / 2$ , and  $r$  be  $n$ .

Note that, in equation (2), the term  $\sum_{i=1}^k \frac{y_i^2}{\delta_i^2}$  describes the projection  $x$  onto the differential latent semantic index space, while  $\epsilon(x)$  approximates the projection of  $x$  on the orthogonal space of the differential latent semantic index space.

## 2.3 Details of Algorithm

### 2.3.1 Setting Up Retrieval System

1. Identify words and noun phrases from stop words.
2. Set up the term list as well as the global weights.
3. Set up the document vectors of all the collected documents in terms of normalized vectors.
4. Set up an interior differential term-document matrix  $D_I^{m \times n_1}$  such that each of its columns is an interior differential document vector.
5. Decompose  $D_I^{m \times n_1}$  by an SVD algorithm such that  $D_I = USV^T$ , then approximate  $D_I$  by  $D_{I,k_1} = U_{k_1} S_{k_1} V_{k_1}^T$  with a proper choice of  $k_1$ . We now define the likelihood function by  $P(x|D_I) =$

$$\frac{n_1^{1/2} \exp\left(-\frac{n_1}{2} \sum_{i=1}^{k_1} \frac{y_i^2}{\delta_i^2}\right) \cdot \exp\left(-\frac{n_1 \epsilon^2(x)}{2\rho_1}\right)}{(2\pi)^{n_1/2} \prod_{i=1}^{k_1} \delta_i \cdot \rho_1^{(r_1-k_1)/2}},$$

where  $y = U_{k_1}^T x$ ,  $\epsilon^2(x) = \|x\|^2 - \sum_{i=1}^{k_1} y_i^2$ ,  $\rho_1 = \frac{1}{r_1-k_1} \sum_{i=k_1+1}^{r_1} \delta_i^2$ ,  $r_1$  is the rank of matrix  $D_I$ . In practical cases,  $\rho_1$  may be chosen as  $\delta_{k_1+1}^2 / 2$ , and  $r_1$  be  $n_1$ .

6. Construct an exterior differential term-document matrix  $D_E^{m \times n_2}$ , such that each of its column is an exterior differential document vector.
7. Decompose  $D_E$  by SVD algorithm, such that  $D_E = USV^T$ , then with a proper value of  $k_2$ , define the  $D_{E,k_2} = U_{k_2} S_{k_2} V_{k_2}^T$  to approximate  $D_E$ . After that, we define the likelihood function  $P(x|D_E) =$

$$\frac{n_2^{1/2} \exp\left(-\frac{n_2}{2} \sum_{i=1}^{k_2} \frac{y_i^2}{\delta_i^2}\right) \cdot \exp\left(-\frac{n_2 \epsilon^2(x)}{2\rho_2}\right)}{(2\pi)^{n_2/2} \prod_{i=1}^{k_2} \delta_i \cdot \rho_2^{(r_2-k_2)/2}},$$

where  $y = U_{k_2}^T x$ ,  $\epsilon^2(x) = \|x\|^2 - \sum_{i=1}^{k_2} y_i^2$ ,  $\rho_2 = \frac{1}{r_2-k_2} \sum_{i=k_2+1}^{r_2} \delta_i^2$ ,  $r_2$  is the rank of matrix  $D_E$ . In practical cases,  $\rho_2$  may be chosen as  $\delta_{k_2+1}^2 / 2$ , and  $r_2$  be  $n_2$ .

8. Define the posteriori function  $P(D_I|x) =$

$$\frac{P(x|D_I)P(D_I)}{P(x|D_I)P(D_I) + P(x|D_E)P(D_E)},$$

where  $P(D_I)$  is set to be an average number of recalls divided by the number of documents in the data base and  $P(D_E)$  is set to be  $1 - P(D_I)$ .

### 2.3.2 Online Document Search

1. A query is treated as a document; a document vector is set up by generating the terms as well as their frequency of occurrence, and thus a normalized document vector is obtained for the query. For each document in the data base, the procedures in item 2-5 are processed.
2. Construct a differential document vector  $x$  using the query.
3. Calculate the interior document likelihood function  $P(x|D_I)$ , and calculate the exterior document likelihood function  $P(x|D_E)$  for the document.
4. Calculate the Bayesian posteriori probability function  $P(D_I|x)$ .
5. Select the documents such that  $P(D_I|x)$  exceeds a given threshold (say, 0.5), or choose the best  $N$  documents with largest  $P(D_I|x)$ , those values of  $P(D_I|x)$  are shown as the scores to rank the match.

## 3. An Example

We demonstrate the power of the current method by some small example below. Suppose we have 4 documents named A, B, C, D at hand. And  $A_1, A_2$  are the abstracts of A obtained by different methods; similarly,  $B_1, B_2, C_1, C_2$  and  $D_1, D_2$  are the abstracts of B, C and D as obtained by different methods respectively. Suppose that  $A_1, A_2, B_1, B_2, C_1, C_2$  and  $D_1, D_2$  are given as follows

$A_1$ : We were successful in our research on the new alloys.

$A_2$ : In studying the novel alloy, they attained success.

$B_1$ : Galileo's research influenced the physical sciences in a big way.

$B_2$ : The physical research was impacted enormously by Galileo's theories.

$C_1$ : Performance was raised as a result of many improvements.

$C_2$ : By making a large number of improvements, its performance was enhanced.

$D_1$ : The objective of this project is to improve performance.

$D_2$ : To improve performance is this project's target.

Since we have only 4 documents in the database in this example representing the documents A, B, C, D respectively, let us arbitrarily save  $A_1, B_2, C_1$  and  $D_2$  in the database. As it is mentioned in section 2, the use of different stemming algorithms differs very little for IR performance. After removing the stop words, Porter's stemming algorithm gives the following stems here: *attain, alloi, enhanc, enorm, Galileo, impact, improv, influenc, novel, object, perform, physic, project, rais,, research, result, science, studi, success, target, theori.*

Each document is associated with two document vectors and we could further normalize each document vector (column) according to equation (1).

We then construct differential term-document matrix  $D_I^{m \times n_1}$  as table 1 (here  $m = 21, n_1 = 4$ )

**Table 1** Interior Differential term-document matrix

	$A_1 - A_2$	$B_1 - B_2$	$C_1 - C_2$	$D_1 - D_2$
attain	-0.447213595	0	0	0
alloi	0.130136674	0	0	0
enhanc	0	0	-0.577350269	0
enorm	0	-0.40824829	0	0
Galileo	0	0.038965305	0	0
impact	0	-0.40824829	0	0
improv	0	0	-0.077350269	0
influenc	0	0.447213595	0	0
novel	-0.447213595	0	0	0
object	0	0	0	0.5
perform	0	0	-0.077350269	0
physic	0	0.038965305	0	0
project	0	0	0	0
rais	0	0	0.5	0
research	0.577350269	0.038965305	0	0
result	0	0	0.5	0
science	0	0.447213595	0	0
studi	-0.447213595	0	0	0
success	0.130136674	0	0	0
target	0	0	0	-0.5
theori	0	-0.40824829	0	0

By decomposing this  $D_I$  by an SVD algorithm, and choosing  $k_1 = 3$ , estimating that  $r_1 = n_1 = 4$ , we have the function  $P(x|D_I)$ :

$$P(x|D_I) = 0.083335295 \times \exp(-2 * (y_1^2/0.9744453796$$

$$+ y_2^2/0.897312874756 + y_3^2/0.845300037604) \times \exp(-4\epsilon^2(x)), \quad (3)$$

where  $y = U_{k_1}^T x, \epsilon^2(x) = \|x\|^2 - (y_1^2 + y_2^2 + y_3^2)$ .

We then construct the exterior differential term-document matrix  $D_E^{m \times n_2}$ ; by choosing  $n_2 = 4$ , we have the matrix of table 2 which of course depends on  $n_2$ .

**Table 2** Exterior Differential term-document matrix

	$A_1 - B_1$	$B_2 - C_2$	$C_1 - D_2$	$D_1 - A_2$
attain	0	0	0	-0.447213595
alloi	0.577350269	0	0	-0.447213595
enhanc	0	-0.577350269	0	0
enorm	0	0.40824829	0	0
Galileo	-0.447213595	0.40824829	0	0
impact	0	0.40824829	0	0
improv	0	-0.577350269	0	0.5
influenc	-0.447213595	0	0	0
novel	0	0	0	-0.447213595
object	0	0	0	0.5
perform	0	-0.577350269	0	0.5
physic	-0.447213595	0.40824829	0	0
project	0	0	-0.5	0.5
rais	0	0	0.5	0
research	0.130136674	0.40824829	0	0
result	0	0	0.5	0
science	-0.447213595	0	0	0
studi	0	0	0	-0.447213595
success	0.577350269	0	0	-0.447213595
target	0	0	-0.5	0
theori	0	0.40824829	0	0

Decomposing this  $D_E$  by SVD algorithm, and choosing  $k_2 = 2$ , estimating that  $r_2 = n_2 = 4$ , we have  $P(x|D_E)$ :

$$P(x|D_E) = 0.023984708 \times \exp(-2(y_1^2/2.6247888144 + y_2^2/2.0277475201)) \times \exp(-2.18448113581\epsilon^2(x)), \quad (4)$$

where  $y = U_{k_2}^T x, \epsilon^2(x) = \|x\|^2 - (y_1^2 + y_2^2)$ .

Suppose the average number of recalls be 1. Then  $P(D_I|x)$  becomes

$$P(D_I|x) = \frac{0.25P(x|D_I)}{0.25P(x|D_I) + 0.75P(x|D_E)}. \quad (5)$$

To select the most likely final candidates to a query  $q$ , we may retain all the documents  $t$  whose threshold  $P(D_I|t - q)$  exceed a certain threshold value or choose the best  $N$  documents  $t$  having the  $N$  largest  $P(D_I|t - q)$  as the candidate group to the query. In this small example, we will choose the document  $t$  with largest  $P(D_I|t - q)$  as the best document to an arbitrary query  $q$ .

Now, suppose we want to search a closest document (documents) to the query:

*The result is influenced by the study of science*

The normalized document vector of this query is:  $q=(0, 0, 0, 0, 0, 0, 0, 0, 0.5, 0, 0, 0, 0, 0, 0, 0.5, 0.5, 0.5, 0, 0, 0)^T$ . For all the documents  $A_1, B_2, C_1$  and  $D_2$ , we can obtain the differential document vectors with the query:

$A_1 - q$ ,  $B_2 - q$ ,  $C_1 - q$  and  $D_2 - q$ .

We calculate the  $D(x|D_I)$ ,  $P(x|D_E)$  and  $P(D_I|x)$  for each of vectors  $A_1 - q$ ,  $B_2 - q$ ,  $C_1 - q$  and  $D_2 - q$ , obtaining final results of  $P(D_I|A_1 - q) = 0.075476859$ ,  $P(D_I|B_2 - q) = 0.155099594$ ,  $P(D_I|C_1 - q) = 0.076703526$  and  $P(D_I|D_2 - q) = 0.029596402$

Choosing the largest one, we see that the document: "The physical research was impacted enormously by Galileo's theories." is the closest to the query. It is most interesting to note that the query "The result is influenced by the study of science" does not share even one common keywords between the query and the document.

#### 4. Conclusion

We have developed a new information retrieval scheme by setting up a most likelihood posteriori function providing a measure of reliability in retrieving a document in the database by a query. Our simple experiment gives a supporting evidence for the validity of the theory, which is capable of capturing the intricate variability in word usage thus improving the instability of traditional keyword-based query search. Since the SVD packages for very large sparse matrixes [3] are now available, and the system set-up step is actually an off-line process, the extension of the computational method to a larger scale IR system should not be difficult.

Because of semantic invariance characteristics of the LSI vector space, we believe that the system can also be used for cross language retrieval. The system should be practical when there is a large set of cross language documents as well as a set of summaries as the training set. As mentioned in the introduction section already, we require each document to be represented by two or more document vectors. The documents vectors can be constructed off line, of course, as long as their summaries are available. Otherwise, we should use some summarization method/software to obtain summaries for each document. We want to emphasize here that while the quality of the summarization is important for our IR system, the summarization document constructed need not be grammatically correct.

#### References

- [1] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391-407, 1990.
- [2] Michael W. Berry, Zlatko Drmac, and Elizabeth R. Jessup, "Matrices, vector spaces, and information retrieval," *SIAM Rev.*, vol. 41, no. 2, pp. 335-362, 1999.
- [3] Michael W. Berry, Susan T. Dumais, and G. W. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Rev.*, vol. 37, pp. 573-595, 1995.

- [4] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery *Numerical Recipes in C: The Art of Scientific Computing*, 2nd edition, Cambridge University Press, Cambridge, England, January 1993.
- [5] V. V. Raghavan and S. K. M. Wong, "A critical analysis of vector space model for information retrieval," *Journal of the American Society for Information Science*, vol. 37, no. 5, pp. 279-87, 1986.
- [6] Hinrich Schütze and Craig Silverstein, "Projections for efficient document clustering," in *Proceedings of SIGIR'97*, 1997, pp. 74-81.
- [7] C. J. van Rijsbergen, *Information retrieval*, Butterworths, 1979.
- [8] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol.14, no. 3, pp. 130-137, 1980.
- [9] M. Lennon, D. Pierce, B. Tarry and P. Willett, "An Evaluation of Some Conflation Algorithms for Information Retrieval", *Journal of Information Science*, vol.3, pp. 177-183, 1981.
- [10] W. B. Frakes, "Stemming Algorithms", in Frakes, William B. and Baeza-Yates, Ricardo (Eds.), *Information Retrieval: Data Structures and Algorithms*, Englewood Cliffs, NJ: Prentice-Hall, 1992. pp. 131-160.
- [11] Gerard Salton, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
- [12] Gerard Salton, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513-524, 1988.
- [13] M. Turk, and A. Pentland, "Eigenfaces for Recognition" *J. of Cognitive Neuroscience*, vol. 3, no.1, pp.71-86, 1991.