Framework for Document Retrieval using Latent Semantic Indexing

Neelam Phadnis
Computer Engineering (M.E)
Thadomal Shahani Engg.College
Mumbai, India

Jayant Gadge
Computer Engineering (M.E)
Thadomal Shahani Engg. College
Mumbai, India

ABSTRACT
Today, with the rapid development of the Internet, textual information is growing rapidly. So document retrieval which aims to find and organize relevant information in text collections is needed. With the availability of large scale inexpensive storage the amount of information stored by organizations will increase. Searching for information and deriving useful facts will become more cumbersome. How to extract a lot of information quickly and effectively has become the focus of current research and hot topics.

The state of the art for traditional IR techniques is to find relevant documents depending on matching words in users' query with individual words in text collections. The problem with Content-based retrieval systems is that documents relevant to a users' query are not retrieved, and many unrelated or irrelevant materials are retrieved. In this paper information retrieval method is proposed based on LSI approach. Latent Semantic Indexing (LSI) model is a concept based retrieval method that exploits the idea of vector space model and singular value decomposition. The goal of this research is to evaluate the applicability of LSI technique for textual document search and retrieval.

General Terms
Information Retrieval

Keywords
Document Retrieval, Latent Semantic Indexing, Singular value decomposition

1. INTRODUCTION
In today's world, with the advent of computers the amount of information stored is growing phenomenally in quantity and variety. This information explosion has resulted in a great demand for efficient and effective means for organizing and indexing data so that useful information can be retrieved whenever required. Thus in order to provide users with easy access to the information in which the user is interested some mechanism is needed. This mechanism should be able to retrieve the relevant text timely and accurately.
Data retrieval, in the context of an information retrieval system, consists mainly of determining which documents of a collection contain the keywords in the user query. In fact the user of an IR system is concerned more with retrieving information about a subject than with retrieving data which satisfies a given query. Users want the retrieval on the basis of conceptual context. A given concept can be exhibited in number of ways (polysemy). So the literal terms in a users query may not match those of relevant documents. Also most words have multiple meanings (synonymy) so terms in a user's query match words in documents that are of no use to the user. A new approach to document retrieval which is designed to overcome the fundamental problem of existing retrieval techniques is presented here.
In this paper, the proposed approach tries to overcome the problems with term matching retrieval. Statistical techniques are used to estimate the hidden latent semantic structure. Latent Semantic Indexing is one such statistical information retrieval technique. It is based on an algebraic model of document retrieval and uses a dimension reduction technique known as Singular Value Decomposition. In these techniques documents are converted into a collection of weighted terms and the goal is to place documents on the same topic close together and dissimilar documents sufficiently apart [1]. Since the search is based on the concepts contained in the documents rather than the documents constituent terms, LSI can retrieve documents related to a users query even when the query and documents do not share any common terms.

2. RELATED WORK
From thousands of years people have practiced the art of archiving and then finding information from this data. The practice of archiving can be traced back to 3000 BC. Even then Sumerians realized the importance of proper organization and access was needed for efficient use of data.

The need to store and retrieve information became more important with inventions like paper and printing. With the advent of computers the amount of data being stored increased dramatically as retrieving information from them could be done mechanically. Vannevar Bush published an article in 1945 that gave birth to the concept of automatic access to large amounts of stored information. Several ideas emerged in the mid 1950's based on searching for text with the help of a computer. Most notable was the development of SMART system by Gerard Salton at Harvard University [2]. The simplest form of document retrieval is linear scan through documents. But it is not efficient when we need to search large document collections quickly.
One common problem with information retrieval systems is the issue of predicting which documents are relevant and which are not. Such a decision is usually dependent on a ranking algorithm which attempts to order the documents. Documents at the top of the ranked list are likely to be more relevant. A ranking algorithm operates according to basic premises regarding the notion of document relevance. Distinct set of premises yield distinct information retrieval models. The three classic models in information retrieval are the Boolean, Probability and Vector space models. An information retrieval model consists of a set of representations

for the documents in the collection and user queries, a framework for modeling their relationships and a ranking function which defines an ordering among the documents with respect to the query. The Boolean model is a simple retrieval model based on set theory and Boolean algebra. Most search engines are Boolean systems that mainly use simple Boolean query operators such as "+" with terms that should be included in retrieved documents and "-" with terms that should be excluded. These systems allowed users to specify their requirement using combination of Boolean ORs, ANDs and NOTs [2]. This model is very easy to understand and implement. But its drawbacks are it only retrieves exact matches. Most users of IR systems expect a ranked document that is the best answer to a query among many documents. Boolean systems do not return ranked documents. Hence it is less effective than ranked retrieval systems.

Today it is well known that index term weighting can lead to substantial improvement in retrieval performance. Index term weighting brings us to the vector space model. In the vector space model developed by Salton in early 70s the document is represented as a set of terms and their associated weight. The weight of the term is the measure of the importance of the term in representing the information contained in the document. These term weights are ultimately used to compute the degree of similarity between each document stored in the system and the user query. The model sorts the documents in decreasing order of this degree of similarity [2]. The resultant answer set is a lot more precise than that retrieved by the Boolean model.

The probabilistic model estimates the probability of a document being relevant to the user. The similarity measure is the ratio between the probability of finding relevant documents to the probability of finding non-relevant documents. The disadvantage of this model is, the need to guess the initial separation of documents and the adoption of the independence of index terms.

In general, the Boolean model is considered to be the weakest model [2]. Experimentally it is shown by Salton and Buckley that the vector space model is expected to outperform the probabilistic model with general collections.

The proposed approach deals with the vocabulary mismatch problem that is, the words used by the user to describe his topic of interest may be different from the words used in the relevant article. This problem is especially important for large databases where the same concept may be expressed in different ways. Consider a fictional matrix of terms by documents.

TABLE I. SAMPLE TERM BY DOCUMENT MATRIX

	Doc1	Doc2	Doc3
Access	x		
Document	x		
Retrieval	x		X
Information		X*	X*
Theory		X	
Database	x		
Indexing	x		
Computer		X*	X*
REL	R		R
Match		M	M

An "R" in the column represents a document relevant to the query. Thus the documents 1 and 3 are relevant. An "M" in the column indicates a perfect match with the query and the document would have been retrieved. Document 1 is a relevant document, which does not contain any words from the query. Hence it is not retrieved by a straightforward keyword retrieval scheme. Document 2 is an unrelated document but it matches terms in the query, and therefore would be returned. But as per the context of the document it should not have been retrieved [3].

To overcome this problem, efforts have been taken to embed semantic information into text processing systems. Other approaches include using a thesaurus for information retrieval. Unfortunately this approach does not work well in general because the relationships captured in a thesaurus frequently are not valid in the local context of a user query. Later approaches implemented grammars and ontologies [4].

Latent Semantic Indexing (LSI) proposed by Deerwester in 1990 is an efficient information retrieval algorithm. LSI model introduces an interesting conceptualization of the information retrieval problem based on the theory of Singular Value Decomposition. The main idea is to map each document and query vector into a low dimensional space which is associated with concepts. The claim is that retrieval in the reduced space may be superior to retrieval in the space of index terms. This results in noise reduction and removal of redundancy. It also deals effectively with sparse, ambiguous and contradictory data. [4] Roslan Sadjirin and Nurrazah Abd Rahman have observed through evaluation of Malay document retrieval, LSI performed 40% better compared to exact term-matching technique. [5]

Weimer Hastings has shown that the true power of LSI comes primarily from the SVD algorithm. Ding has constructed a statistical model for LSI using the cosine similarity measure. This paper describes the LSI process for document retrieval using SVD and interprets the resulting matrices in a geometric context.

3. PROPOSED APPROACH

In this paper, information retrieval method is proposed based on Latent Semantic Indexing approach. LSI model is a concept based retrieval method that exploits the idea of vector space model and singular value decomposition.

3.1 Latent Semantic Indexing

Latent Semantic Indexing (LSI) is an approach based on the Vector Space Model. In LSI, we perform an approximation of a term-document matrix C by one of low rank using singular value decomposition. The low-rank approximation to C yields a new representation for each document in the collection. Queries are also cast into this low-rank approximation enabling us to compute document similarity scores. This process is known as latent semantic indexing.

The general idea is to map documents, the terms in the documents to a low dimensional representation. This low dimensional space reflects semantic associations between the terms. The document similarity is computed based on the inner product in this latent space. As a result, queries against a set of documents that have undergone LSI will return results that are conceptually similar in meaning to the query even if they don't share a specific word or words with the query. Thus LSI addresses the problem of synonymy. In this paper, in order to improve the retrieval effectiveness, a concept based search on a corpus is performed using LSI [6].

3.2 Vector Space Model

The Vector Space Model was invented by Salton and his colleagues for the SMART information retrieval system and is the most commonly used model in Document Retrieval

system today. In VSM model a document is represented as a point in space. Points that are close together are semantically similar. Query is represented as a point in the same space as the documents. Thus a collection of N documents can be viewed as a collection of vectors leading to a term-document matrix M*N whose rows represent the M terms of the N columns each of which corresponds to a document [7]. To compensate for the effect of the length of a document the standard way of quantifying the similarity between two documents is to compute the cosine similarity of their vector representation. The vector inner product is also often used as the similarity measure. The advantage of this approach is that VSMs extract information automatically from a corpus, thus they require much less labor than the other approaches to semantics such as thesaurus and ontologies.

3.2.1 Term-document representation:

The effective retrieval of relevant information is directly affected by the logical view of the documents adopted by the retrieval system. Documents in a collection are frequently represented through a set of index terms or keywords. These keywords provide a logical view of the document. The full text is clearly the most complete logical view of a document. With very large collections we need to reduce the set of representative keywords. This can be accomplished through the elimination of stopwords and the use of stemming. This reduces the complexity of the document representation resulting in a set of index terms.

3.3 Singular Value Decomposition

The basic idea behind Singular Value Decomposition is taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data. The dimensionality reduction helps in omitting the details in the document. It selects the least costly mapping and maps synonyms to the same dimension.

3.3.1 Computing the SVD:

SVD is based on a theorem in linear algebra which states that a rectangular matrix A can be decomposed into a product of three other matrices- an orthogonal matrix U, a diagonal matrix Σ and the transpose of an orthogonal matrix V such that when the three components are matrix-multiplied, the original matrix is reconstructed.

$A = U\Sigma V^T$ ... (1)

Where $U^TU=I$, $V^TV=I$, the columns of U are orthonormal eigenvectors of AA^T , the columns of V are orthonormal eigenvectors of A^TA and Σ is a diagonal matrix containing the square roots of eigenvalues from U or V in descending order.

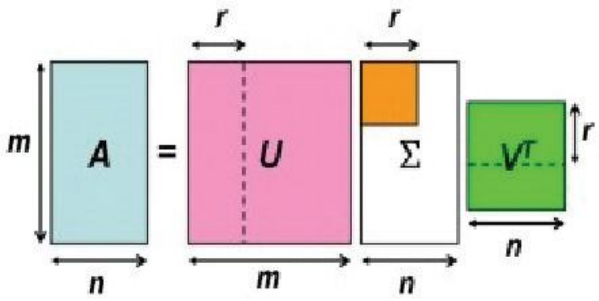

Fig.1: Singular Value Decomposition [7]

The first k columns of U and V matrices and first k singular values of A [8] are used to construct a rank k approximation to A via

$A_k = U^k \Sigma^k V_k^T$ .... (2)

Thus, for IR purposes, SVD provides a reduced model for representing the *term-to-term*, *document-to-document* and *term-to-document* relationships [1].

Even for a collection of medium size the term document matrix is likely to have several tens of thousands of rows and columns and a rank in the ten of thousand as well. In latent semantic indexing, singular value decomposition is used to construct a low rank approximation C_K for a value of K that is far smaller than the original rank of C. It is generally chosen to be in the low hundreds. Each row and column is thus mapped to a K dimensional space.

Next this k dimensional LSI representation is used to compute similarities between vectors. The fidelity of the approximation of C_K to C leads us to hope that the relative values of cosine similarities are preserved, that is if a query is close to a document in the original space it remains relatively close in the k-dimensional space.

The proposed approach consists of two parts, a preprocessing phase where the term document matrix is built and the query phase in which the user query is also converted into a similar representation. A useful document is expected to share many features with the query. Similarity between each document and the query is computed and the most relevant documents are returned to the user.

3.4 Preprocessing the Dataset

The following sections present the pre-processing of the input dataset, which is necessary before performing LSI.

3.4.1 Tokenization

The main objective of Tokenization is treating digits, hyphens, punctuation marks and the case of letters. Numbers are not good index terms because without a surrounding context they are inherently vague. Breaking up hyphenated words is useful. Punctuation marks are removed entirely in this step. The case of letters is usually not important for the identification of index terms. Hence we convert all text to either lower or upper case.

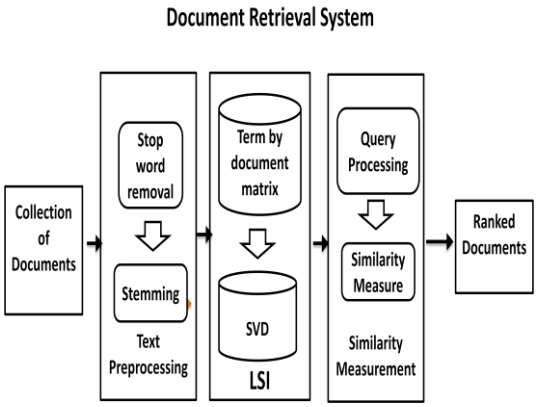

Fig.2 : Framework for Document Retrieval System

3.4.2 Stop Word Removal

Words which are too frequent among the documents in the collection are not good discriminators. A word which occurs in 80% of the documents is useless for the purpose of retrieval. Such words are referred to as stopwords. (e.g. Articles {a, an, the} Prepositions {at, by, in, to, from} Conjunction {and, but, as} others {become, everywhere}). They are not considered influential during the execution of LSI process to retrieve relevant documents. It also reduces the size of the indexing structure considerably [9].

3.4.3 Stemming

Morphology is the study of internal structure of words. Stemming is a kind of morphological analysis to reduce a word to its stem or root form. Often words are composed of stem with added affixes such as plural forms and past tenses. Users searching for information on 'retrieval' will also be interested in texts containing information about retrieve, retrieving etc. Stemming improves retrieval performance by reducing distinct index terms. Porter's stemming algorithm is used for this purpose [10].

3.4.4 Dictionary

To compare one document to another the documents must be represented by vectors. The dictionary is a vector that contains all the unique words of all documents in the document set. The value of each dimension in a document's vector is the frequency of a specific word in that document [11].

The output of this pre-processing step is the input of the LSI retrieval system to generate the term by document matrix.

3.5 LSI Retrieval System

3.5.1 Latent Semantic Indexing Algorithm:

The following algorithm is used to perform LSI on the test collection.

1. Convert each document from test collection to a list of words (terms) into a vector of word.

2. Scale each vector of terms so that every term reflects the frequency of its occurrence in the document.

3. Combine these column vectors into a large term document matrix. Rows represent terms and columns represent documents.

4. Perform SVD on the term document matrix. This will result in three matrices commonly called U, Σ and V.

5. Set all singular values to 0 except the k highest singular values.

6. Recombine the terms U, eigenvector of matrix Σ and document (V^T), to form original matrix $U\Sigma V^T = A$.

7. Break this reduced rank term-document matrix back into column vectors. Associate these with their respective documents [4].

3.5.2 Term-document Matrix

In this step the text is represented as a matrix in which each row stands for a unique word and each column stands for a document. Each cell contains the frequency with which the word of its row appears in the document denoted by its column. This kind of matrix is called a term-document matrix. Let A be a term-document matrix,

$$A = [a_{ij}] \ldots (1)$$

Where a_{ij} indicates not only that term i occurs in document j but also the number of times the term appears in that document. [9]. After building the term-document matrix Normalize it using

$$B(i,j) = \frac{A_{(i,j)}}{n} \ldots (2)$$

where A is the TDM and n is the total no of words in document j. Apply local and global weight transformation on the normalized TDM. Local weight

$$L_{(i,j)} = tf_{(i,j)} \ldots (3)$$

where $tf_{(i,j)}$---normalized frequency of term i in document j

Global weight idf or G(i,j)

$$G_{(i,j)} = \log\left(\frac{d}{dfi}\right) \ldots (4)$$

where d is the number of documents in the whole collection and dfi be the frequency of documents in which term i occurs.

Thus the weight,

$$w_{i,j} = tf_{(i,j)} * \log\left(\frac{d}{dfi}\right) \ldots (5)$$

3.5.3 Query Projection and Matching

In the LSI model queries are formed into pseudo-documents that specify the location of the query in the reduced term-document space.

$$q = q^T U_k \Sigma^{-1}_k \ldots (8)$$

Thus the pseudo document consists of sum of the term vectors

($q^T U_k$) corresponding to the terms specified in the query scaled by the inverse of the singular values(Σ^{-1}_k) [1].

For the query term weights

$$W_{i,q} = \left(0.5 + 0.5 \frac{freq_{i,q}}{maxi\ freq_{\ i,q}}\right) * \log\left(\frac{N}{n_i}\right) \ldots (9)$$

where $freq_{i,q}$ is the raw frequency of the term k_i in the query. Once the query is projected in the term document space, one of the similarity measures is applied to compare the position of the pseudo-document to the position of the terms or documents in the reduced term-document space.

3.5.4 Similarity Measurement:

The query vector q is defined as $q=(w_{1,q}, w_{2,q}, \ldots, w_{t,q})$ where t is the total number of index terms in the system. The document vector is represented as $d_j=(w_{1,j}, w_{2,j}, \ldots, w_{t,j})$. The vector model evaluates the degree of similarity of the

document d_j with regard to the query q as correlation between vectors d_j and q.

$$Sim(d_j, q) = \frac{\left(\sum_{i=1}^{t} w_{i,j} * w_{i,q}\right)}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} * \sqrt{\sum_{i=1}^{t} w_{i,q}^2}} \ldots (10)$$

Since wi,j ≥ 0 and wi,q ≥ 0, sim(q,dj) varies from 0 to 1. Thus instead of attempting to predict whether a document is relevant or not, the vector model ranks the documents according to their degree of similarity to the query [12].

3.5.5 Evaluation Method:

Evaluation of the LSI technique uses well-known IR measures recall and precision. Recall is used to measure the relevant documents which are effectively retrieved. On the other hand precision is used to measure the retrieved documents known to be relevant. If the recall is 1, it means that all the relevant documents are retrieved though there could be retrieved documents that are not relevant. If the precision is 1, it means that the entire retrieved documents are relevant, though there could be relevant documents that were not retrieved [1].

$$Recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents} \ldots (11)$$

$$Precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ retrieved\ documents} \ldots (12)$$

4. CONCLUSION

In this paper, an approach to improve the efficiency of document retrieval using LSI is examined. The LSI approach is useful in finding textual information in large collections compared to traditional retrieval techniques which ignore the underlying semantic structure of data. It overcomes vocabulary problems (synonymy) by automatically extracting the semantic structure. It involves the use of singular value decomposition to build a matrix whose cells represent the co-occurrence of terms within documents so that it could identify terms and documents which are similar. This methodology was used to rank text documents in response to user query. In future this approach can be combined with Wordnet to improve web information retrieval.

5. REFERENCES

[1] Todd A Letsche, Micheal W Berry. “Large Scale Information Retrieval with Latent Semantic Indexing”. Information Sciences 1997.

[2] Singhal, Amit. "Modern information retrieval: A brief overview." *IEEE Data Eng. Bull.* 24.4 (2001): 35-43.

[3] Deerwester, Scott C., et al. "Indexing by latent semantic analysis." *JASIS* 41.6 (1990): 391-407.

[4] Roger Bradford. “Why LSI? Latent Semantic Indexing and Information Retrieval**”** 2009 Content Analyst Company.

[5] Sadjirin, Roslan, and Nurazzah Abd Rahman. "Efficient retrieval of Malay language documents using Latent Semantic Indexing." *Information Technology (ITSim), 2010 International Symposium in*. Vol. 3. IEEE, 2010.

[6] Aswani Kumar, Ch, and Suripeddi Srinivas. "Latent semantic indexing using eigenvalue analysis for efficient information retrieval." *International Journal of Applied Mathematics and Computer Science* 16 (2006): 551-558.

[7] Rodrigues, Ravina, and Kavita Asnani. "Concept based search using LSI and automatic keyphrase extraction." *Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference on*. IEEE, 2010.

[8] Yang, Jianxiong, and Junzo Watada. "Decomposition of term-document matrix representation for clustering analysis." *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*. IEEE, 2011.

[9] Zaman, A. N. K., and Charles Grant Brown. "Latent semantic indexing and large dataset: Study of term-weighting schemes." *Digital Information Management (ICDIM), 2010 Fifth International Conference on*. IEEE, 2010.

[10] Porter, Martin F. "An algorithm for suffix stripping." *Program: electronic library and information systems* 14.3 (1980): 130-137.

[11] Symeonidis, Panagiotis, Ivaylo Kehayov, and Yannis Manolopoulos. "Text classification by aggregation of SVD eigenvectors." *Advances in Databases and Information Systems*. Springer Berlin Heidelberg, 2012.

[12] Zhao, Rong, and William I. Grosky. "Narrowing the semantic gap-improved text-based web document retrieval using visual features." *Multimedia, IEEE Transactions on* 4.2 (2002): 189-200.

[13] Berry, Michael W., Susan T. Dumais, and Todd A. Letsche. "Computational methods for intelligent information access." *Supercomputing, 1995. Proceedings of the IEEE/ACM SC95 Conference*. IEEE, 1995.

[14] Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. "An introduction to latent semantic analysis." *Discourse processes* 25.2-3 (1998): 259-284.

[15] Furnas, George W., et al. "Information retrieval using a singular value decomposition model of latent semantic structure." *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1988.

[16] Dumais, Susan T., et al. "Using latent semantic analysis to improve access to textual information." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1988.