Latent Semantic Analysis: Searching Technique for Text Documents

[1]Rajandeep Kaur, [2]Manpreet Kaur
[1,2]Assistant Professor
[1]Department of Information Technology, [2] Department of Computer Science
[1]Sant Baba Bhag Singh Institute of Engineering & Technology
[1]Padhiana, Jalandhar (Punjab), India

***Abstract*—The paper describes about the searching in text document with its concepts. The searching of text relevant document with the help of keyword or query is known as information retrieval. The information retrieval technique which is used to improve indexing of stored documents and search engine query performance is known as LSA (Latent Semantic Analysis) is explained in this paper. LSA, the technique of searching documents, is also known as LSI (Latent Semantic Indexing). LSA involves a matrix operation called singular value decomposition, which is main component of analysis of LSA.**

***Index Terms*—Text Document, latent semantic analysis, latent semantic indexing, precision, recall, frequency table**

I. INTRODUCTION

A text is any sequence of symbols (or characters) drawn from an alphabet. A large portion of the information available worldwide in electronic form which is actually in text form (other popular forms are structured and multimedia information). Some examples of such kind of text documents are books, journals, articles, newspapers, jurisprudence databases, corporate information, and the Web. A text database is a system that maintains a large text collection and provides fast and accurate access to it [1]. Textual data appear in an ever-increasing number of business and research situations, and are encountered by Information Systems (IS) researchers in a variety of contexts. For example, researchers in the IS discipline have an interest in examining titles, abstracts, or full-text bodies of IS publications in order to identify attributes such as research topics, theories, and methods, related to the nature of the research [2]. The searching in text documents can be performed through two methods:

- Semantic search: A text search on natural language text based on the meaning rather than the syntax of the text. [3]
- Syntactic search: A text search based on string patterns that appear in the text, without reference to meaning. [4]

II. BASIC CONCEPTS OF SEARCHING IN TEXT DOCUMENTS

Precision and Recall

An issue related to relevance computation is how to evaluate the accurateness of the results. The most popular measures are called precision and recall. In information retrieval contexts, precision and recall are defined in terms of a set of retrieved documents and a set of relevant documents. Precision is the percentage of relevant retrieved documents out of the total number of documents retrieved by the system on a query. Recall is the percentage of relevant retrieved documents out of all relevant documents. [5]

$$\text{Precision} = \frac{\text{No. of relevant documents retrieved}}{\text{Total no. of documents retrieved from the file}}$$

$$\text{Recall} = \frac{\text{No. of relevant documents retrieved}}{\text{Total no. of relevant documents in the file}}$$

The probabilistic interpretation of precision states that precision is the probability that a (randomly selected) retrieved document is relevant. The probabilistic interpretation of recall states that recall is the probability that a (randomly selected) relevant document is retrieved in a search. [6]

Stop List and Word Stem

Stop list is a set of words that do not "discriminate" between the documents in a given archive. Word Stem as many words are small syntactic variants of each other. E.g., drug, drugged, drugs are similar in the sense that they share a common "stem," the word drug. Most document retrieval systems first eliminate words on stop lists and reduce words to their stems, before creating a frequency table. [7]

Frequency Table

If D is a set of N documents and T is a set of M words/terms occurring in the documents of D and Assume no words on the stop list for D occur in T and all words in T have been stemmed.

Then, The frequency table FreqT is an (M×N) matrix such that FreqT(i,j) equals the number of occurrences of the word t_i in the document d_j.
For example, we have 3 documents with doc id d1, d2, d3 as shown in Table 1. The frequency table for same is given in Table 2 which is made with word stem & after removing stop list from given document strings.

Table 1 Documents with their string data

Doc	String
d1	Drug Connections
d2	Boats and Drugs
d3	Connections between Terrorism and Drugs

Table 2 Frequency Table for given documents

Term/Doc	d1	d2	d3
Drug	1	1	1
Connection	1	0	1
Boat	0	1	0
Terror	0	0	1

III. LATENT SEMANTIC ANALYSIS

Latent Semantic Analysis (LSA) is a theory and method for extracting and representing the contextual-usage meaning of words by statistical computations applied to a large corpus of text [8]. They asserted that LSA could serve as a model for the human acquisition of knowledge. From the original application for retrieving information, the use of LSA has evolved to systems that more fully exploit its ability to extract and represent meaning. LSA is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. It is not a traditional natural language processing or artificial intelligence program; it uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, or morphologies, or the like, and takes as its input only raw text parsed into words defined as unique character strings and separated into meaningful passages or samples such as sentences or paragraphs. [9]
LSA uses 4 steps approach which is explained as follow:

- Table creation: The creation of the frequency matrix (table) FreqT is performed in first step.
- SVD Construction: The computation of the singular valued decompositions (A,S,B) of FreqT is done in this step.
- Vector Identification: For each document d, let vec(d) be the set of all terms in FreqT whose corresponding rows have not been eliminated in the singular matrix S
- Index Creation: Store the set of all vec(d)'s indexed by any one of the number of techniques (such as TV-tree).

Table creation

The first step of LSA is to create frequency table for given document d. The creation of a term-by document matrix is done where the columns are documents and the rows are terms [10]. A term is a subdivision of a document; it can be a word or phrase or some other unit. A document can be a sentence, a paragraph, a textbook, or some other unit. In other words, documents contain terms. The elements of the matrix are weighted word counts which represents how many times each term appears in each document.

Singular Value Decomposition

LSA involves a matrix operation called singular value decomposition. A singular value decomposition of FreqT is a triple (A, S, B) where:

- FreqT = $(A \times S \times B^T)$
- A is an (M × M) orthogonal matrix such that $A^TA = I$
- B is an (N × N) orthogonal matrix such that $B^TB = I$
- S is a diagonal matrix called a singular matrix [12]

The Eq. 1 and Eq. 2 are used for computing the values of A, S and B which are given below:

$$C^TC = AS^TSB^T \quad (1)$$
$$CB = AS \quad (2)$$

where C is representing Frequency Table (i.e. FreqT).
Given a frequency matrix FreqT, we can decompose it into SVD TSD^T where S is non-decreasing
If FreqT is of size (M × N), then T is of size (M × M) and S is of order (M × R) where R is the rank of FreqT, and D^T is of the order (R × N). Usually the value of R is taken as 200.

The "magic" performed by LSA is to reduce S, the diagonal matrix created by SVD, to an appropriate number of dimensions resulting in S'. The product of TSD^T is the least-squares best fit to X, the original matrix [10].

Vector Identification

We can now shrink the problem substantially by eliminating the least significant singular values from the singular matrix S

- Choose an integer k that is substantially smaller than R
- Replace S by S^*, which is a (k × k) matrix such that $S^*(i,j) = S(i,j)$ for $1 \leq i, j \leq k$
- Replace the (R × N) matrix D^T by the (k × N) matrix D^{*T} where $D^{*T}(i,j) = D^T(i,j)$ if $1 \leq i \leq k$ and $1 \leq j \leq N$

For shrinking the matrix, throw away the least significant values and retain the rest of the matrix. Key claim in LSI is that if k is chosen judiciously, then the k rows appearing in the singular matrix S^* represent the k "most important" (from the point of view of retrieval) terms occurring in the "entire" document. Finding the correct number of dimensions for the new matrix created by SVD is critical; if it is too small, the structure of the data is not captured. Conversely, if it is too large, sampling error and unimportant details remain. [11]

Index Creation

Telescopic vector Tree (TV) is used for index creation for high dimensional data. It organizes data in hierarchical structure. The objects are at leaf node and their description these objects are stored at parent node. Parent nodes are recursively grouped too, until the root is formed. As more objects are inserted into the tree, more features might be needed to discriminate among the objects. The key point is that feature nodes are introduced whenever needed. [12]

IV. ADVANTAGES AND DISADVANTAGES

The advantages and disadvantages of LSA are given as follow [13]:

Advantages

- In LSA, the concept in question, as well as all documents, that are related to it is all likely to be represented by a similar manner.
- LSA analysis recovers the original semantic structure of the space and its original dimensions. The new dimensions by LSA analysis are a better representation of documents and queries.
- By using a reduced representation in LSA, also help to remove some "noise" from the data. The noise is a data which could be described as rare and less important usages of certain terms.
- LSA factors are orthogonal by definition; hence data is positioned in the reduced space in a way that reflects the correlations in their use across documents and helps in better retrieval.

Disadvantages

- LSA vectors require large storage. There are many advances in electronic storage media, but still the loss of sparseness due to large data is more serious implication.
- LSA performs relatively well for long documents due to the small number of context vectors used to describe each document. However, due to large size of data it requires an additional storage space and computing time which reduces the efficiency of LSA.
- Another objection to SVD is that, it is designed for normally-distributed data, but such a distribution is inappropriate for all type of data.

V. APPLICATIONS OF LSA

The several promising applications of LSA are given below [14]:

- LSA can be used for information retrieval. The LSA woks better than other methods such as standard vector methods when the queries and relevant documents do not share many words.
- Researchers are also carrying out some interesting work of LSA in medical field.
- LSA can be used for information filtering.
- LSA can also be used to return the best matching people instead of document, where people were represented by articles they had written.
- LSA does not depend on literal keyword matching, so it is useful when the text input is noisy, as in OCR (Optical Character Reader), open input, or spelling errors.
- LSA can be used as electronic feedback or e- assessment for e- learning. [11]
- LSA can be used to word sense discrimination within a tutor for English vocabulary learning. [15]

VI. CONCLUSION

LSA is good method for searching text documents i.e. for information retrieval & in many other applications. The LSA can be considered as a two dimensional representation of the documents and question vectors. Singular value decomposition and dimension reduction of SVD result play an important role in the performance. But as we have studied in disadvantages, LSA is relatively difficult to apply to large amount of data because SVD needs more computation & also more space is needed to save data which may lead to low performance.

ACKNOWLEDGEMENT

We would like to thank all the authors who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. I sincerely thank to my parents, family, and friends, who provide the advice and financial support. The product of this research paper would not be possible without all of them.

REFERENCES

[1] Gonzalo Navarro, "Text Document", Encyclopedia of Database Technologies and Applications, Idea Group Inc., Pennsylvania, USA. ISBN 1-59140-560-2, pages 688-694, 2005.
[2] Nicholas Evangelopoulos, Xiaoni Zhang, Victor R. Prybutok, "Latent Semantic Analysis: five methodological recommendations", European Journal of Information Systems (2012) 21, 70–86.
[3] G. Navarro, R. Baeza-Yates, E. Sutinen, and J. Tarhio. "Indexing methods for approximate string matching". IEEE Data Engineering Bulletin, 24(4):19–27, 2001.
[4] G. Navarro and M. Raffinot. "Flexible Pattern Matching in Strings – Practical on-line search algorithms for texts and biological sequences" Cambridge University Press, 2002.
[5] Erzsebet Toth, Bela Lorant Kovacs, "Technical relevance of keyword searches in full text databases", Qualitative and Quantitative Methods in Libraries (QQML) 2:477 –484, 2014
[6] William H. Walters, "Comparative Recall and Precision of Simple and Expert Searches in Google Scholar and Eight Other Databases", portal: Libraries and the Academy, Vol. 11, No. 4 (2011), pp. 971–1006.
[7] A N K Zaman , "Stop Word Lists in Document Retrieval Using Latent Semantic Indexing: an Evaluation" Journal of E-Technology, Volume 3 Number 1 February 2012.
[8] Landauer, T. K. & Dumais, S. T. , " A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge", 104, 211-140, 1997
[9] D. Hull, "Improving text retrieval for the routing problem using Latent Semantic Indexing", in Proceedings of the Seventeenth Annual International ACM-SIGIR Conference, 1994.
[10] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," Journal of the American Society for Information Science, vol. 41, pp.391-407, 1990.
[11] Debra Trusso Haley, Pete Thomas, Bashar Nuseibeh, Josie Taylor, Paul Lefrere "The Learning Grid and E-Assessment using Latent Semantic Analysis", Computing Research Centre, The Open University, 2005.
[12] King-lp Lin, H.V. Jagadish, and Christos Faloutsos, "The W-Tree: An Index Structure for High-Dimensional Data", VLDB Journal,3, 517-542, 1994.
[13] Alan Kaylor Cline, Inderjit S. Dhillon "Computation of the Singular Value Decomposition" available www.cs.utexas.edu/users/inderjit/public_ papers /HLA_SVD.pdf
[14] Barbara Rosario, "Latent Semantic Indexing: An overview", INFOSYS 240 Spring 2000.
[15] Juan Pino and Maxine Eskenazi, "An Application of Latent Semantic Analysis to Word Sense Discrimination for Words with Related and Unrelated Meanings", NAACL HLT Workshop on Innovative Use of NLP for Building Educational Applications, pages 43–46, Boulder, Colorado, June 2009.