International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869, Volume-2, Issue-10, October 2014

Latent Semantic Analysis for Information Retrieval

Khyati Pawde, Niharika Purbey, Shreya Gangan, Lakshmi Kurup

Abstract—**This paper presents a statistical method for analysis and processing of text using a technique called Latent Semantic Analysis. Latent semantic analysis was a technique that was devised to mimic human understanding of words and language. Hence it is a method for computer simulation of the meaning of word and passages by analysis of natural language or text. It uses a mathematical model called Singular Value Decomposition which is a technique used to factorize a matrix. The paper discusses its application in information retrieval, which is called latent semantic indexing in this context. We also present an example which demonstrates this technique.**

***Index Terms*—Information Retrieval, Latent Semantic Analysis, Latent Semantic Indexing, Singular Value Decomposition.**

I. INTRODUCTION

In artificial intelligence, developing algorithms that can automatically process natural language and text has been a big challenge. The demand for computer systems to manage and filter search through huge repositories has increased to a great extent over the years. This paper presents an approach called latent semantic analysis (LSA) which is a method in natural language processing for extracting and representing contextual-usage meaning of words by statistical computations that is applied to a large amount of text [1]. Latent semantic analysis examines the relationship between a set of documents and terms and after processing a large sample of data, it represents the words used in the document in a high dimensional semantic space [3].

While a lot of statistical methods like vector space model, probabilistic model and document clustering can be used for information retrieval, our paper concentrates on the latent semantic indexing methodology. Latent semantic indexing is an information retrieval technique which indexes and identifies the pattern in unstructured collection of text and the relationship between them [2]. It uses a mathematical technique called singular value decomposition (SVD) to identify the relationships. This paper includes a detailed description of the entire latent semantic indexing process [7]. Latent semantic indexing generates associations between the terms that occur in similar context. It is based on the principle that the words used in the same context tend to have similar meanings. We analyze how effective LSI is and how SVD can be improved [2]. With our paper's main focus being on information retrieval using LSI, we analyze the pros and cons of this technique with respect to information retrieval.

The paper is organized as follows. Section 2 of the paper introduces the different components of the process and the process of LSA. It also consists of an illustrative example relevant to the technique. Section 3 offers an improvement to the SVD technique and pertinent suggestions are given. In Section 4, we discuss why LSI is an appropriate method to be used for the purpose of information retrieval. Finally, Section 5 presents our conclusion.

Manuscript received October 23, 2014.

Khyati Pawde, Niharika Purbey, Shreya Gangan, Lakshmi Kurup, Computer Engineering,, Dwarkadas J Sanghvi College of Engineering, Mumbai, India.

II. LATENT SEMANTIC ANALYSIS

Latent semantic analysis is a technique which represents the meaning of a word as a kind of average of the meaning of all the documents in which it appears, and the meaning of the document as a kind of average of the meaning of all the word that the document contains [1].

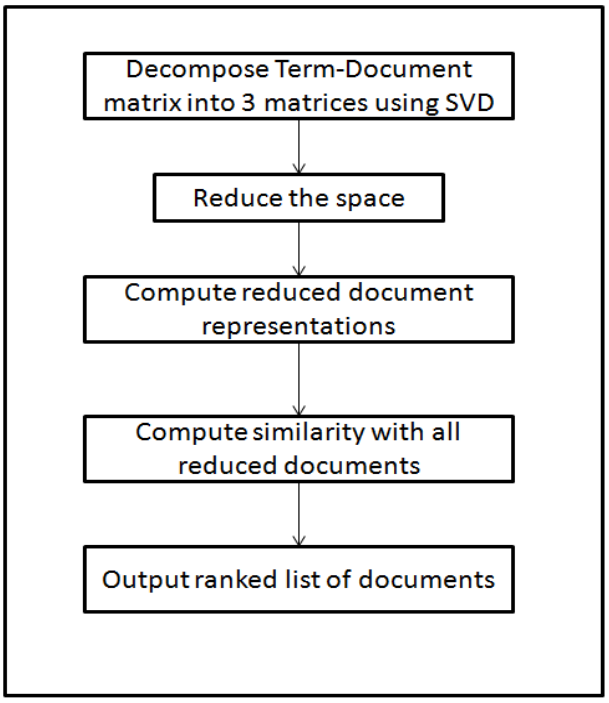

Fig. 1. Flowchart representing the steps involved in LSA.

A. Count Data and Co-occurrence Matrix

LSA uses count data which is a statistical data type, wherein the observations can take only non-negative values and these values are a result of counting. For example, we consider a set of text documents D with the terms from a universal vocabulary W. The data collected is organized into a co-occurrence matrix with a count which would denote the number of occurrences of different words [4]. This matrix is

also called as term-document matrix where the values denote the number of times a particular term occurs in the document.

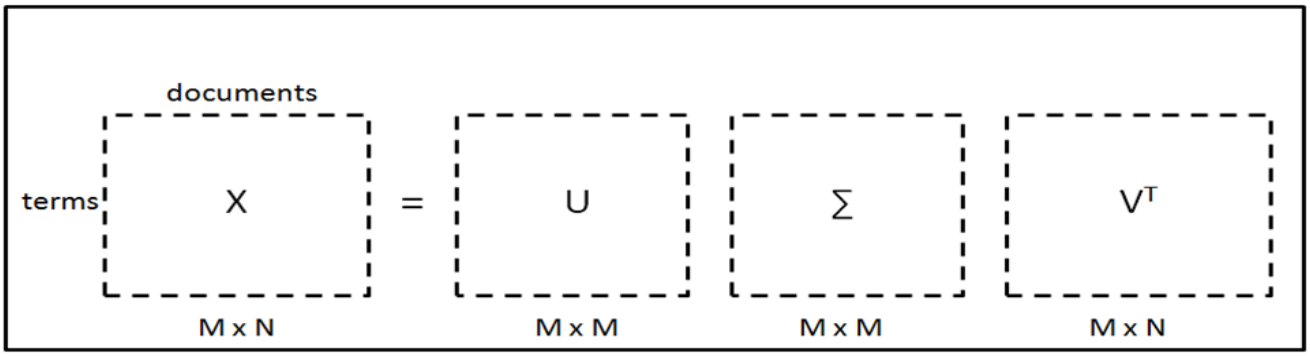

Fig. 2. Singular Value Decomposition of the term- document matrix X^2

Data sparseness is a major issue with the co-occurrence matrix and it is immediately identified. This problem is also known as zero-frequency problem. A typical matrix may contain very few of the non-zero entries, which means that very few of the words in the vocabulary are actually used in a single document [5]. The problem here is that, for an application where queries are matched with documents or where similarities in documents are evaluated by comparing the common terms, the likelihood to find the common terms even in closely related articles may be small. This is because the terms might not be exactly the same.

Many of the matching functions that are based on similarity factors take the inner product between the pairs of the document vectors. The problem faced can be seen in two ways. First, synonyms need to be accounted for so that true similarity between the documents can be generated. But at the same time, polysemy also has to be dealt with in order to avoid overestimating the similarity between the documents which count common terms with different meanings. Polysemy is the ability of a word to have multiple associated meanings. Both of these issues may result in incorrect matching score which may not reveal the actual similarity between the documents.

B. Rank Lowering

After constructing the co-occurrence table, a low rank approximation is constructed. A low rank approximation is a problem, wherein the cost function measures an agreement between the data matrix and an approximating matrix which is an optimization variable, where the approximating matrix has a reduced rank. There many reasons for building a low rank approximation. First, since the term-document matrix is very large. It would eliminate the problem of huge size, noisy and overly sparse term-document matrix. It would also to some extent mitigate the problem of polysemy.

C. Latent Semantic Analysis by SVD

The matrix constructed previously is analyzed by Singular Value Decomposition to derive particular latent semantic structure model. The singular value decomposition (SVD) is the factorization of complex or real matrix. The decomposition provides a reduced rank approximation in column and row space of the document matrix. The analysis begins with the matrix of associations between all pairs of one type of object.

The process of decomposition is called Eigen-analysis and it results into two matrices of special form. These matrices show a breakdown of the original data into linearly independent components [6][9].

The decomposition is defined as $X=U\sum V^T$ where, columns of U are orthogonal eigenvectors of XX^T, columns of V are eigenvectors of X^TX[10]. Each of the original documents' similarity behavior is approximated by the corresponding values in the smaller number of factors. The result can be geometrically represented by spatial configuration, wherein the dot product or cosine of the angle between the vectors representing the two documents corresponds to the estimated similarity.

Let us consider an example to illustrate this process.
Consider the original matrix X as follows:

Table 1: Original matrix X

X	D1	D2	D3	D4	D5	D6
airplane	1	0	1	0	0	0
flight	0	1	0	0	0	0
air	1	1	0	0	0	0
plastic	1	0	0	1	1	0
tree	0	0	0	1	0	1

This is a standard term-document matrix which is non-weighted.

We then construct matrix U as follows:

Table 2: Matrix U

U	1	2	3	4	5
airplane	-0.44	-0.30	0.57	0.58	0.25
flight	-0.13	-0.33	-0.59	0.00	0.73
air	-0.48	-0.51	-0.37	0.00	-0.61
plastic	-0.70	0.35	0.15	-0.58	0.16
tree	-0.26	0.65	-0.41	0.58	-0.09

This matrix represents one row per term and one column per min(M,N) where M is the number of terms and N is the number of documents.

Each value U_{ij} in matrix indicates how strongly a term i is related to the topic represented by semantic dimension j.

We then construct the ∑ matrix as follows:

Table 3: Matrix ∑

∑	1	2	3	4	5
airplane	2.16	0.00	0.00	0.00	0.00
flight	0.00	1.59	0.00	0.00	0.00
air	0.00	0.00	1.28	0.00	0.00
plastic	0.00	0.00	0.00	1.00	0.00
tree	0.00	0.00	0.00	0.00	0.39

This is a square, diagonal matrix of dimensionality min(M,N) × min(M,N). The diagonal consists of the singular values of C. The magnitude of the singular value measures the importance of the corresponding semantic dimension.

We then construct the V^T matrix as follows:

Table 4: Matrix V^T

V^T	D1	D2	D3	D4	D5	D6
1	-0.75	-0.28	-0.20	-0.45	-0.33	-0.12
2	-0.29	-0.53	-0.19	0.63	0.22	0.41
3	0.28	-0.75	0.45	-0.20	0.12	-0.33
4	0.00	0.00	0.58	0.00	-0.58	0.58
5	-0.53	0.29	0.63	0.19	0.41	-0.22

This matrix represents one column per document and one row per min(M,N) where M is the number of terms and N is the number of documents. Each value V_{ij} in the matrix indicates how strongly document i is related to the topic represented by semantic dimension j.

Thus we have decomposed the term-document matrix into a product of three matrices. Further processing of these matrices can be done for the purpose of dimensionality reduction.
By "zeroing out" all but the two largest singular values of ∑, we obtain $∑_2$ as,

Table 5: Matrix $∑_2$

$∑_2$	1	2	3	4	5
airplane	2.16	0.00	0.00	0.00	0.00
flight	0.00	1.59	0.00	0.00	0.00
air	0.00	0.00	0.00	0.00	0.00
plastic	0.00	0.00	0.00	0.00	0.00
tree	0.00	0.00	0.00	0.00	0.00

We can compute X_2 from this as,

Table 6: Matrix X_2

X_2	D1	D2	D3	D4	D5	D6
1	-1.62	-0.60	-0.44	-0.97	-0.70	-0.26
2	-0.46	-0.84	-0.30	1.00	0.35	0.65
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

We notice that the last three rows of each of the matrices $∑_2$ and X_2 are populated by zeros. Hence, the SVD product can be carried out using only two rows in the representations of $∑_2$ and V^T. Therefore, we can replace these matrices by their truncated versions $∑'_2$ and $(V')^T$ as:

Table 7: Matrix $(V')^T$

$(V')^T$	D1	D2	D3	D4	D5	D6
1	-1.62	-0.60	-0.44	-0.97	-0.70	-0.26
2	-0.46	-0.84	-0.30	1.00	0.35	0.65

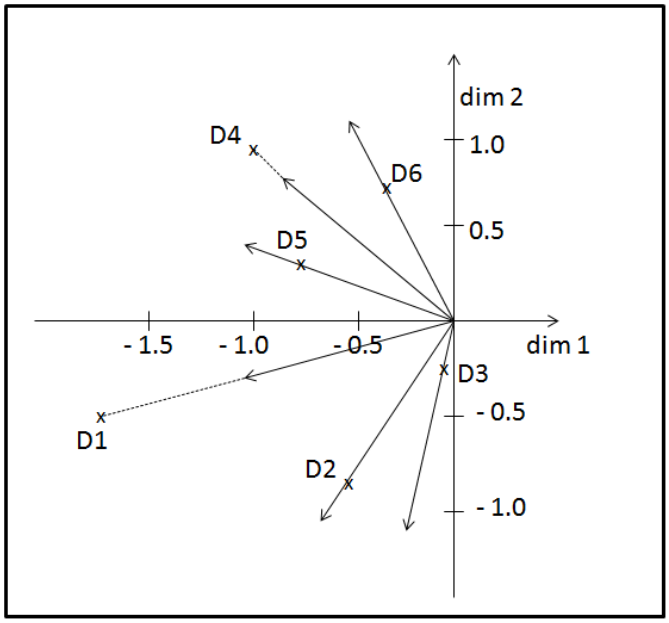

Fig. 3. The documents of the above example reduced to two dimensions

I. OPTIMIZATION OF SVD

The SVD can offer optimal results only when no other matrix which is of the same rank or has the same underlying dimensionality, can approximate X better [8].

Upon analysis, it can be inferred that Eckart-Young Low Rank Approximation Theorem is one of the most effective techniques used to increase the optimality of SVD. This theorem gives an optimal approximation of the original matrix X by retaining the k largest singular values and setting all the other values to zero. The measure of this approximation can be traced by the Frobenius norm which is

$$\|X\|_F = \sqrt{(\sum_i \sum_j x_{ij}^2)} \quad (1)$$

II. INFORMATION RETRIEVAL

The foremost advantage of using the LSI methodology is that it takes documents which are semantically similar but are not analogous in the vector space. LSI represents these documents in a reduced vector space which in turn elevates their degree of similarity. The dimensionality reduction makes the procedure neglect all the details in the document. The cost of collapsing unrelated words is much more than mapping synonyms to the same dimension. In such a way, LSI deals with the problems of synonymy and semantic relatedness which occur in the process of information retrieval. Hence, LSI is the preferred method for Information Retrieval in which it correctly matches queries to documents of similar topical meaning when the query and documents use different words [11].

For the purpose of information retrieval, the user's query is represented as a vector in k dimensional space and compared to each of the documents. The implementation involves computing the SVD of the term-document matrix, reduction of the space and computation of the reduced document representation, mapping the query onto the reduced space which is $q_2^T=\sum_2^{-1} U_2^T q^T$ which follows from $X_2=U\sum_2 V^T$ i.e. $\sum_2^{-1}U^T=V_2^T$ and finally estimation of the similarity of q_2 with all reduced documents in V_2.

Generally, relevance feedback and query expansion are used in increase recall in information retrieval if the query and the documents have no common terms. LSI is also used to increase the recall and in turn could hurt the precision. Thus, we can say that it address the same problems as the previously used methods.

III. CONCLUSION

It is evident from our research that LSA is used for information retrieval since it tackles the problems of synonymy in which the same underlying concept is described using different terms, polysemy where each word could have more than one meaning, and term dependence as in the association between correlated terms across different documents making it much superior to other traditional retrieval strategies. This technique also has certain drawbacks which include large storage requirements and a high computing time which reduces efficiency. Ultimately, to decide if the advantages outweigh the disadvantages, the retrieval performance needs to be taken into consideration. LSA provides better results as compared to the plain vector model. There are a few other techniques such as Probabilistic Latent Semantic Indexing and Latent Dirichlet Allocation which eliminate certain flaws of LSA. The results of the LSA do not introduce well defined probabilities and are hence difficult to interpret. The Probabilistic LSA tackles this problem and provides a sound statistical foundation for analysis offers better model selection and reduces complexity. The major advantage of using a model like LDA is that it can be scaled up to provide useful inferential machinery in domains involving multiple levels of structure [13]. But the LSA being a very popular method, which has already been tried on diverse datasets makes it extremely reliable. Thus, we can conclude that though LSA lacks important cognitive abilities that humans use to construct and apply knowledge from experience, the success of LSA as a theory of human knowledge acquisition and representation should not be underestimated.

ACKNOWLEDGMENT

We would like to extend our gratitude to our honorable principal Dr. Hari Vasudevan of D.J. Sanghvi College of Engineering and Dr.Narendra Shekhokar, the Head of Department of Computer Engineering for granting us the required amenities for our research.

REFERENCES

[1] Landauer, T. K., Foltz, P. W., & Laham, D., Introduction to Latent Semantic Analysis. Discourse Processes, 1998, 25, 259-284.
[2] S. Deerwester, S. T. Dumais, G. W. Furnas, Landauer. T. K., and R. Harshman. Indexing by latent semantic analysis, Journal of the American Society for Information Science, 41, 1990.
[3] T. K. Landauer and S. T. Dumais, "A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge," Psychological Review, vol. 104, pp. 211-240, 1997.
[4] P.W. Foltz, Using Latent Semantic Indexing for information filtering, COCS '90 Proceedings of the ACM SIGOIS and IEEE CS TC-OA conference on Office Information systems, pg 40-47.
[5] Thomas Hofmann, Probabilistic latent semantic analysis, UAI'99 Proceedings of the Fifteenth conference on uncertainty in artificial intelligence, pg. 289-296.
[6] T. Landauer, D. McNamara, S. Dennis, and W. Kintsch, Handbook of Latent Semantic Analysis. Lawrence Erlbaum Associates, 2007.
[7] S.T. Dumais, G.W. Furnas, T.K. Landauer, S. Deerwester, R. Harshman, Using Latent Semantic Analysis to improve access to textual
information, In proceedings of the SIGCHI conference on human factors in computing systems, p.281-285.
[8] Amudaria S, Sasirekha S, Improving the precision ration using semantic based search, 2011 International Conference on Computing and Networking Technologies, IEEE, 465-470.
[9] Zongli Jiang, Changdong Lu, A Latent Semantic Analysis Based Method of Getting the Category Attribute of Words, 2009 Electronic Computer Technology International Conference, IEEE, 141-146.
[10 Jing Gao, Jun Zhang Clustered SVD strategies in latent semantic indexing, USA Oct 2004.
[11] Baker, F.B. Information retrieval based on latent class analysis. Journal of the ACM, 1962, 512-521.
[12] An Introduction to Information Retrieval, Stanford NLP Online Book, 2009.
[13] David B., Andrew Ng, Michael J., Latent Dirichlet Allocation, Journal of Machine Learning Research, 2003, 993-1022.

Khyati Pawde, B.E. in Computer Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai, India.

Niharika Purbey, B.E. in Computer Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai, India.

Shreya Gangan, B.E. in Computer Engineering, Dwarkadas J Sanghvi College of Engineering, Mumbai, India.

Lakshmi D. Kurup, Assistant Professor (Computer Engineering Department), Dwarkadas J Sanghvi College of Engineering, Mumbai, India.