ISSN:0975-9646

Improved Similarity Based Matrix Completion Algorithm Using Latent Semantic Indexing

Navneet kaur Waraich
Computer Science Department
Punjab Technical University
Punjab, India

Hardeep Singh Sindher
Computer Science Department
Punjab Technical University
Punjab, India

***Abstract*—LSI usually is conducted by using the singular value decomposition (SVD). The main difficulty in using this technique is its retrieval performance depends strongly on the choosing of an appropriate decomposition rank. In this paper, by observing the fact that the SVD makes the related documents more connected, we devise a improved matrix completion algorithm. The proposed algorithm returns results that are meaningful to the search criteria. Latent Semantic Indexing (LSI) helps in information filtering where certain type of words are removed from retrieved documents and indexing is performed to bring the essential results at the top and so on according to the defined algorithm.**

I. INTRODUCTION

Latent semantic indexing offers the ability to associate semantically related terms in the set of files. Latent semantic indexing permit observing semantics in the set of files and how meaning of the text in the documents is extracted. This is used to get a reply to queries that return more "significant" results and not just a keyword search.

Latent semantic analysis (LSA) find out the underlying latent semantic structure of the words usage in a body of text and how the meaning of the text in reaction to user queries is taken out, commonly referred to as concept searches.

LSI conquers two of the most difficult restrictions of Boolean or keyword queries: First is synonymy in which multiple words have similar meanings and second is polysemy in which words have more than one meaning. Synonymy is often the cause of mismatches in the vocabulary utilized by the users of information retrieval systems. As an outcome, Boolean or keyword queries often return inappropriate results and miss information that is relevant.

LSA provides a process for determining the resemblance of significance of words and passages by analysis of large text amount. For effectiveness wording does not require to be in sentence form for latent semantic indexing. It also works with free-form notes, email, lists, web based content and so on. Presented that a collection of text contains many terms, latent semantic indexing can be used to spot patterns in the association between the important expressions and concepts enclosed in the text.

II. RELATED WORK

Golub et al. [16] described a numerically stable and fairly fast scheme to compute the unitary matrices U and V which transform a given matrix A into a diagonal form Σ. Deerwester et al. [11**]** described the automatic indexing and retrieval that took benefit of implicit higher-order structure in the association of terms with documents in order to improve the discovery of appropriate documents on the basis of terms found in queries. Golub et al. [17] represented essential information about the mathematical background and algorithmic skills required for the production of numerical software. Kolda et al. [19] represented the semi-discrete decomposition (SDD) LSI method and compared it with SVD (Singular value decomposition) LSI method. This paper updated the SDD for a dynamically changing document collection. Lee et al. [22] described two issues pertaining to URIs, and presented recommendations. Section 1 addressed how URI space was separated and the connection between URIs, URLs, and URNs. Section 2 described how URI schemes and URN namespace ids were registered. Dhillon [12] described the suggestion of modelling the document collection as a bipartite graph between documents and words, using which the simultaneous clustering trouble was posed as a bipartite graph partitioning problem.

Guha et al. [18] described an application called Semantic Search which provided an outline of TAP, an application framework upon which the Semantic Search is built. Drineas et al. [13] represented a paper describing a problem that was solved by computing the Singular Value Decomposition (SVD) of the matrix that represented the m point's; this solution can be used to get a 2-approximation algorithm. Srinivasan et al. [24] described a difficulty of growing availability of web services demands for a discovery mechanism to locate services that satisfy the requirement. Bray et al. [7] represented Extensible Markup Language (XML) as a subset of SGML completely describing this document. Bechhofer et al. [2] described OWL, the Web Ontology Language being designed by the W3C Web Ontology Working Group, enclosed a high-level abstract syntax for both OWL DL and OWL Lite, sublanguages of OWL. Fallside et al. [15] represented XML schema which defined services for defining datatypes to be used in XML Schemas in addition to other XML specifications.

Berry et al. [5] represented a valuable, generally non-technical, insight into how search engines worked, how to improve the users' success in Information Retrieval (IR), and an in-depth analysis of a mathematical algorithm for improving a search engine's performance. Kontostathis et al. [20] described a theoretical model for understanding the performance of latent semantic indexing search and retrieval application. Landauer et al. [21] represented a theory and technique for extracting and representing the contextual-usage importance of expression by statistical calculation applied to a large amount of text. Belew [4] described the idea of Finding out About (FOA), the process of actively seeking out information appropriate to a topic of concern. Bray et al. [6] described XML namespaces providing a simple procedure for qualifying element and attribute names used in extensible mark-up language documents by connecting them with namespaces identified by URI references. Alhabashneh et al. [1] represented an integrated framework for enhancing enterprise search. Tabassum et al. [25] described Relation Based Page Rank Algorithm that relied on information extracted from user queries and relevance is scored on probability.

Crain et al. [10] described two forms of dimension reduction: Latent semantic indexing and topic modelling, together with probabilistic latent semantic indexing and latent Dirichlet allocation. Evangelopoulos et al. [14] represented the influx in generation, storage, and availability of textual information. Preethi et al. [23] described the practice of finding appropriate web pages for any known query from a collection of documents. Thorleuchter et al. [26] represented a multilevel security (MLS) specifically created to protect information from unauthorized access. Connolly et al. [9] represented DAML and OIL which is a semantic markup language for web resources. DAML and OIL provided modelling primitives commonly found in frame-based languages. Beckett [3] represented the Resource Description Framework (RDF) which is a general-purpose language for representing information in the Web. Brickley et al. [8] described how to use RDF to describe RDF vocabularies.

III. INFORMATION RETRIEVAL

The main focal point of information retrieval is to find significant keywords with the aim to classify them. Information retrieval is the activity of acquiring information resources related to an information need. Exploration can be based on metadata or on full text indexing procedure.

When a user penetrates a query into the system then information retrieval begins. Queries are prescribed statements of information needs. A query does not solely identify a single thing in the collection. As an alternative, numerous objects match the query, possibly with different degrees of relevancy.

A. Classification of IR-models

The models are considered according to two proportions: the mathematical foundation and the properties of the model.

1) First proportion: mathematical foundation
   - Set-theoretic models are the models that signify documents as sets of words or phrases. Comparison is generally derived from set-theoretic operations. Common models are:-
     - Extended Boolean model
     - Fuzzy retrieval
     - Standard Boolean model
   - Algebraic models characterize documents and queries typically as vectors, matrices, or tuples. Scalar value is represented by the resemblance of the query vector and document vector. Models under this category are:-
     - Vector space model
     - Generalized vector space model
     - Latent semantic indexing or latent semantic analysis
     - Extended Boolean model
     - Topic-based Vector Space Model
   - Probabilistic models take care of the process of document recovery as a probabilistic knowledge. Connections are figured out as probabilities that a document is appropriate for a given query in the document. Probabilistic theorems like the Bayes' theorem are frequently utilized in these models.
     - Uncertain inference
     - Binary Independence Model
     - Language models
     - Latent Dirichlet allocation
     - Divergence from randomness model
     - Probabilistic relevance model
   - Feature based retrieval models observe documents as assessment of feature functions and seek the finest means to unite these elements into a single relevance score. Feature functions are arbitrary functions of document and query and as such can effortlessly include almost any other retrieval model.

2) Second proportion: properties of the model
   - Models with no term interdependencies treat dissimilar terms/words as independent. The piece of information is generally signified in vector space models by the orthogonality assumption of term vectors or in probabilistic models by an independency assumption for term variables.
   - Models with immanent term interdependencies agree to a representation of interdependencies between terms. It is straightforwardly or indirectly derives from the co-occurrence of those terms in the entire set of documents.
   - Models through transcendent term interdependencies allocate a representation of interdependencies between terms, but they do not declare how the interdependency between two terms is identified. They impart an external source for the level of interdependency between two terms.

IV. SINGULAR VALUE DECOMPOSITION

The SVD is a matrix decomposition procedure that factorizes a rectangular real or complex matrix into its left singular vectors, right singular vectors and singular values. The singular value decomposition (SVD) provides useful applications in different fields.

Singular value decomposition (SVD) is disintegration and is used to decrease the number of elements used to represent the documents. Eigenvector analysis is an efficient means to characterize the correlation structure among huge sets of objects and SVD is one of the techniques used to achieve this.

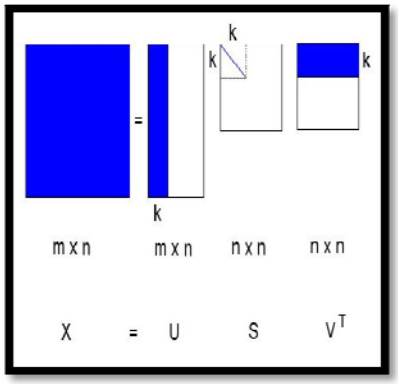

Figure 1: Representation of SVD

Singular value decomposition divides any rectangular matrix of size mxn into three modules: - a mxn matrix (U), a nxn matrix (V^T) and a nxn diagonal matrix (S) which explains the connection between the mxn matrix and the nxn matrix. This is done using the formula:

$$X=USV^T \quad \text{(Eq. 1)}$$

The mxn matrix (U) is created of columns called left singular vectors symbolized as (u_k). The rows of the nxn matrix (V^T) hold the component of the right singular vectors, (v_k). The diagonal matrix(S) include the singular values which are the elements contained by the matrix on the diagonal. The diagonal values are non-zero, although all other elements within the matrix are zero. This means as follows:

$$S=\text{diag}\,(S_1, \ldots\ldots, S_n) \quad \text{(Eq. 2)}$$

The singular vectors are structured by sorting them from elevated to near the ground which denotes the highest singular value is in the upper left index of the diagonal matrix. SVD allows the computation of:

$$X^{(i)} = \Sigma^{i}_{k=1} u_k s_k v^{T}_k \quad \text{(Eq. 3)}$$

A. Synonymy

The words which have the same or almost the same meaning are known as Synonyms. Many words in English have the same or almost the same meaning, for example words in {university, college, institute}, {female, girl, woman}, and {book, novel, biography} are synonyms.

LSI using the improved SVD can recognize synonyms as long as there is a short path that chains the synonyms together. For example: 'mark' and 'twain' are connected to 'Samuel' and 'Clemens' through Doc2. So it can be expected that LSI using the improved SVD is able to recognize the synonyms. Similarly, 'color' and 'purple' are connected through Doc4.

B. Polysemy

Polysemy is the problem of a word with multiple meanings but is not necessarily related. Since a polyseme can appear in unrelated documents, a query containing it will probably also retrieve unrelated documents. For example: problem where 'bank' either refers to financial institution or area near river. If query containing 'bank' and 'money' is made to this vector space model, then only Doc1 and Doc3 will be recognized as relevant since the other documents have the same score. Similarly, if query containing 'river' and 'bank' is made, then only Doc2 and Doc4 will be retrieved.

V. PROPOSED ALGORITHM

The proposed algorithm is designed by explicitly measuring similarities between word pairs to create more connected clusters. The following describes the algorithm.

Let $A \in \mathbb{R}_+^{M*N}$ be the word-by-document matrix. By using cosine criterion, similarity between word p and q can be computed by:

$$s_{pq} = \frac{\mathbf{a}_{p:}\mathbf{a}_{q:}^T}{\|\mathbf{a}_{p:}\|_2 \|\mathbf{a}_{q:}\|_2} \quad \text{(Eq. 4)}$$

where ax: denotes x-th row of A and $s_{pq} \in [0, 1]$. If spq→1, then p and q are strongly related since they co-appear in many documents, and if spq→0 then p and q are unrelated. The proposed algorithm works by propagating entry weights of word vectors to each other based on similarity measures between the vectors. The following update rule is used to update entry j of word i: aij←max(aij, sikakj) $\forall k \neq i$. As shown, if word i and k are related, i.e., sik>0, then the rule will make word i and k co-appear in documents that index either i or k. Algorithm 1 outlines the proposed algorithm, where A(0) denotes the initial matrix (the original word-by-document matrix), a(n) ij denotes (i, j) entry of A at n-th iteration, and maxiter denotes maximum number of iteration. Because the algorithm replaces some zero entries with positive numbers as the update process progresses, we name it as 'Improved Similarity-based Matrix Completion Algorithm'.

Algorithm 1: Improved Similarity-based matrix completion algorithm.

1) Input: $A^{(0)} \in \mathbb{R}^{M\times N}_+$

2) Construct word similarity matrix $S \in \mathbb{R}^{M\times M}_+$ which entry spq is computed using equation.

3) Update entry aij using the following procedure:

for n = 1, . . . , maxiter do

a(n) ij←max(a(n−1), sika(n−1) kj), ∀, j, k ≠ i
end for

VI. Experimental Results

We now evaluate LSI capability of the proposed algorithm numerically using four standard datasets in LSI research i.e. Medline, Cranfield, CISI and ADI [27]. The figure below shows the Frobenius norms per iteration upon the datasets:

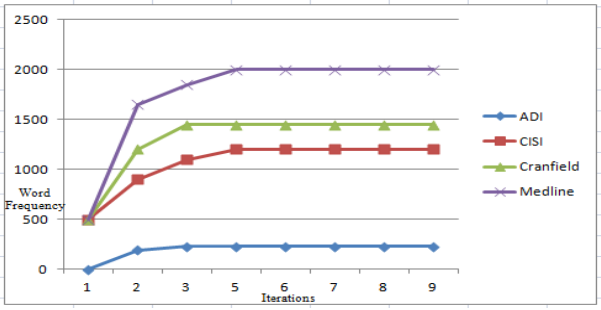

Figure 2: Frobenius norms per iteration

Recall and precision are the most commonly used metrics to measure IR performance.

- Recall measures proportion of retrieved relevant documents to all relevant documents in the collection.

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

- Precision measures proportion of retrieved relevant documents to all retrieved documents.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

The table represents the word frequencies of each dataset upon different iterations.

Table 1: Displaying word frequencies of datasets

Iterations	ADI	CISI	Cranfield	Medline
1	0	500	500	500
2	190	900	1200	1650
3	230	1100	1450	1850
5	230	1200	1450	2000
6	230	1200	1450	2000
7	230	1200	1450	2000
8	230	1200	1450	2000
9	230	1200	1450	2000

A. *Comparison with Medline dataset*

The proposed algorithm results are much more useful as compared to old scheme because it shows precision value of 0.37 which is less than 0.49 precision value. This improves the time dimensionality and results are obtained more quickly. The figure represents the comparison with Medline dataset of old scheme with the new scheme:

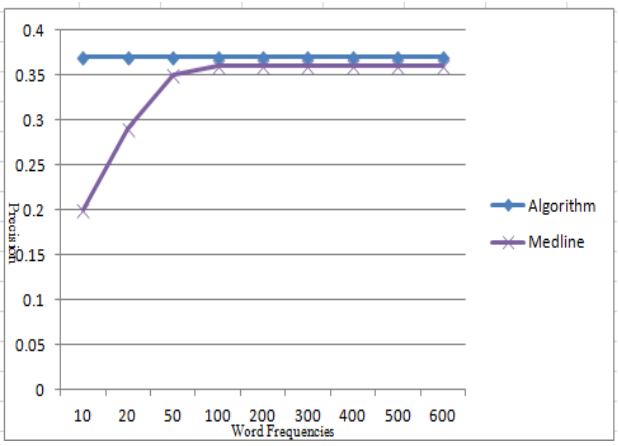

Figure 3: Comparison of algorithm with Medline

Table 2: Results of Comparison with Medline

Word Frequencies	Algorithm	Medline
10	0.37	0.2
20	0.37	0.29
50	0.37	0.35
100	0.37	0.36
200	0.37	0.36
300	0.37	0.36
400	0.37	0.36
500	0.37	0.36
600	0.37	0.36

Improved algorithm of latent semantic analysis has resulted in more efficient manner as compared to previous algorithm.

B. *Comparison with Cranfield dataset*

The proposed scheme results with the old scheme shows a difference of 0.1 precision value which means that the improved version of algorithm is more effective than the old scheme. The figure represents the comparison with Cranfield dataset of old scheme with the new scheme:

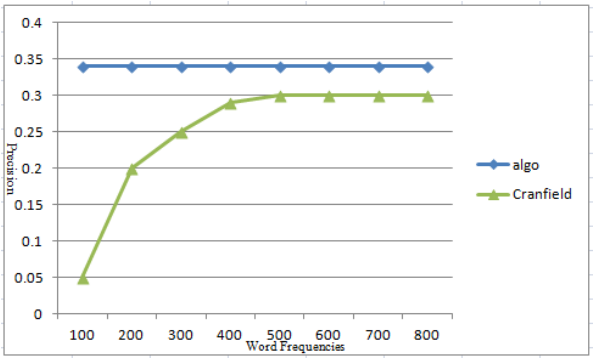

Figure 4: Comparison of algorithm with Cranfield

The results of proposed scheme show precision value of 0.34 as compared to 0.35 precision value of old scheme.

Table 3: Results of Comparison with Cranfield

Word Frequencies	Algorithm	Cranfield
100	0.34	0.05
200	0.34	0.2
300	0.34	0.25
400	0.34	0.29
500	0.34	0.3
600	0.34	0.3
700	0.34	0.3
800	0.34	0.3

C. Comparison with CISI dataset

The comparison between the old and new scheme's precision value is one and a half times less than the precision value of old scheme. The figure represents the comparison with CISI dataset of old scheme with the new scheme:

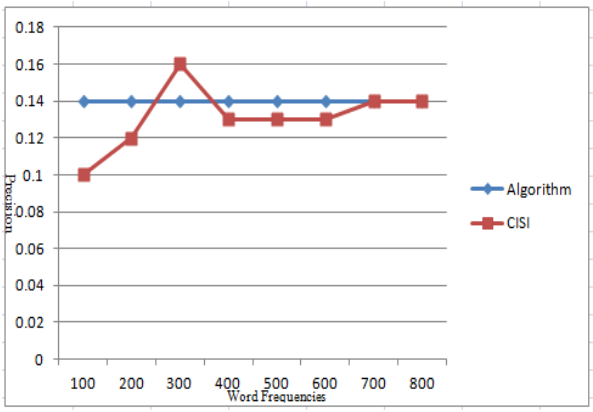

Figure 5: Comparison of algorithm with CISI

The results show 0.14 value for the proposed algorithm as compared to the old scheme upon CISI dataset.

Table 4: Results of Comparison with CISI

Word Frequencies	Algorithm	CISI
100	0.14	0.1
200	0.14	0.12
300	0.14	0.16
400	0.14	0.13
500	0.14	0.13
600	0.14	0.13
700	0.14	0.14
800	0.14	0.14

D. Comparison with ADI dataset

The improvised algorithm presents precision value of two units less than the old scheme. The figure represents the comparison with ADI dataset of old scheme with the new scheme:

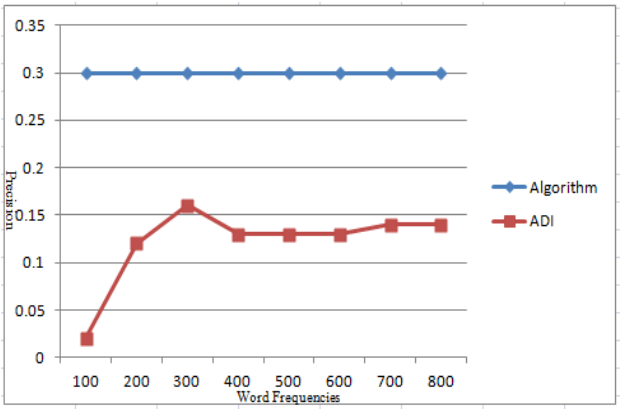

Figure 6: Comparison of algorithm with ADI

The results of the new scheme show precision value of 0.3, whereas, the old scheme shows precision value of 0.32.

Table 5: Results of Comparison with ADI

Word Frequencies	Algorithm	ADI
100	0.3	0.02
200	0.3	0.12
300	0.3	0.16
400	0.3	0.13
500	0.3	0.13
600	0.3	0.13
700	0.3	0.14
800	0.3	0.14

VII. CONCLUSION

The practical tests proved that our method outperforms the other examined methods. The proposed algorithm partitions a heterogeneous collection of information entity with respect to the conceptual domains found and with the help of latent semantic indexing indexes the content of each partitioned subset.

LSI is performed on the datasets to hold data related to the user query. In this manner LSI is applied to datasets that created scalability troubles. Moreover, the computation of the singular value decomposition of the term by document matrix is also accomplished at various dispersed computers increasing the strength of the retrieval systems while decreasing search times.

Latent Semantic Indexing (LSI) is an information retrieval technology which exploits dependencies or "semantic similarity" between terms. Final goal is to create an integrated natural language processing system capable of searching and presenting web documents in a concise and coherent form.

REFERENCES

1. Alhabashneh, O., Iqbal, R., Shah, N., Amin, S. and James, A. (2011), *"Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing"*, Proceedings of the 19th international conference on Conceptual structures, pp. 346–352, 2011.
2. Bechhofer, S., Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. and Stein, L.A. (2004), *"OWL Web Ontology Language"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2004/REC-owl-ref-20040210/>, 2004.
3. Beckett, D. (2014), *"RDF/XML Syntax Specification (Revised)"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2014/REC-rdf-schema-20140225/>, 2014.
4. Belew, R.K. (2008), *"Finding Out About: Cognitive Perspective on search engine technology and the WWW"*, Cambridge University Press, vol.1, 2008.
5. Berry, M.W. and Browne, M. (2005), *"Understanding Search Engines: Mathematical Modelling and Text Retrieval"*, SIAM, Philadelphia, 2005.
6. Bray, T., Hollander, D. and Layman, A. (2009), *"Namespaces in XML"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>, 2009.
7. Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E. and Yergeau, F. (2004), *"Extensible Markup Language (XML) 1.0 (Third Edition)"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2004/RECxml-20040204/>, 2004.
8. Brickley, D. and Guha, R. (2014), *"RDF Vocabulary Description Language 1.0: RDF Schema"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2014/REC-rdf-schema-20140225/>, 2014.
9. Connolly, D., Harmelen, F., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. and Stein, L.A. (2013), *"DAML and OIL Reference Description"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2013/NOTE-daml+oil-reference-20011218/>, 2013.
10. Crain, S.P., Zhou, K., Yang, S.H. and Zha, H. (2012), *"Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond"*, Mining Text Information, pp. 129–161, 2012.
11. Deerwester, S., Dumais, S., Furnas, G., Landauer, T. and Harshman, R. (1990), *"Indexing by latent semantic analysis"*, Journal of the American Society for Information Science, vol. 41, no. 6, pp. 391–407, 1990.
12. Dhillon, I. (2001), *"Co-clustering documents and words using bipartite spectral graph partitioning"*, Proceedings of 7th ACM SIGKDD International Conference, pp. 269–274, 2001.
13. Drineas, P., Frieze, A., Kannan, R., Vempala, S. and Vinay, V. (2004), *"Clustering large graphs via the singular value decomposition"*, Machine Learning, vol. 56, no. 1-3, pp. 9–33, 2004.
14. Evangelopoulos, N., Zhang, X. and Prybutok, V. (2012), *"Latent semantic analysis: Five methodological recommendations"*, European Journal of Information Systems, 21 pp. 70–86, 2012.
15. Fallside, D.C. and Walmsley, P. (2004), *"XML Schema Part 0: Primer Second Edition"*, World Wide Web Consortium, viewed in June 2014, <http://www.w3.org/TR/2004/REC-xmlschema-0-20041028/>, 2004.
16. Golub, G. and Kahan, W. (1965), *"Calculating the singular values and pseudo inverse of a matrix"*, Journal of SIAM Numerical Analysis, vol. 2, no. 2, pp. 205–224, 1965.
17. Golub, G. and Loan, C.V. (1996), *"Matrix computations (3rd edition)"*, Johns Hopkins University Press, pp. 694, 1996.
18. Guha, R., McCool, R. and Miller, E. (2003), *"Semantic Search"*, International Conference on World Wide Web, 2003.
19. Kolda, T. and O'Leary, D. (1998), *"A semidiscrete matrix decomposition for latent semantic indexing information retrieval"*, ACM Transactions on Information Systems, vol. 16, no. 4, pp. 322–346, 1998.
20. Kontostathis, A. and Pottenger, W. (2006), *"A framework for understanding latent semantic indexing (LSI) performance"*, Information Processing and Management, vol. 42, no. 1, pp. 56–73, 2006.
21. Landauer, T.K., Foltz, P.W. and Laham, D. (2007), *"Introduction to Latent Semantic Analysis"*, Psychology Press, vol.25, Issue 2-3, 2007.
22. Lee, T.B., Fielding, R. and Masinter, L. (2001), *"Uniform Resource Identifiers (URI): Generic Syntax"*, World Wide Web Consortium, 2001.
23. Preethi, N. and Devi, Dr.T. (2012), *"Case and relation (CARE) based Page rank Algorithm for Semantic Web Search Engines"*, International Journal of Computer Issues (IJCSI), vol.9, Issue 3, 2012.
24. Srinivasan, N., Paolucci, M. and Sycara, K. (2004), *"An Efficient Algorithm for OWL-S based Semantic Search in UDDI"*, International Semantic Web Services and Web Process Composition Workshop (SWSWPC), vol.3387, 2004.
25. Tabassum, G. and Poongodai, A. (2011), *"An Ontology Based Search for Relevant pages using Semantic Web, Search Engines"*, International Journal of Advanced Engineering Sciences and Technologies (IJAEST), vol.11, Issue 1, April 2011.
26. Thorleuchter, D. and Poel, D.V. (2012), *"Improved multilevel security with latent semantic indexing"*, Expert Systems with Applications, vol. 39, no. 18, pp. 462-471, 2012.
27. Information retrieval group, <http://ir.dcs.gla.ac.uk/resources/test_collections/>, accessed on 15 July 2014.
28. Mirzal, A. (2013), *"Similarity based matrix completion algorithm"*, International Conference on Control System, Computing and Engineering, Dec 2013.