New York Science Journal

Websites:
http://www.sciencepub.net/newyork
http://www.sciencepub.net

Emails:
newyorksci@gmail.com
editor@sciencepub.net

MARSLAND PRESS
Multidisciplinary Academic Journal Publisher

# Comparison of LSI Algorithms without and with Pre-Processing: Using Text Document Based Search

Sheikh Muhammad Saqib

Institute of Computing and information Technology, Gomal University, Dera Ismail Khan, Pakistan
saqibsheikh4@hotmail.com

**Abstract:** Searching of documents/text is the most important need of each student or user computer. Searching through particular index or term is the old fashion, now a day's user want to search documents according to some phrase, query or requirement i.e. extraction of meaningful information from large collection according to some textual query. Different methods such as Iterative Residual Rescaling (IRR), Term Frequency (TF), Inverse Document Frequency (IDF), multi words is using to handle such issues. Latent Semantic Indexing (LSI) is an important method for current literature of information retrieval. LSI can find similar documents on particular textual phrase. Here authors has implemented two algorithms (without and with Pre-Processing) of LSI for text documents. As a result, both algorithms can obtain the similar results but their processing time will be different.
[Sheikh Muhammad Saqib. **Comparison of LSI Algorithms without and with Pre-Processing: Using Text Document Based Search.** *N Y Sci J* 2021;14(1):38-46]. ISSN 1554-0200 (print); ISSN 2375-723X (online). http://www.sciencepub.net/newyork. 7. doi:10.7537/marsnys140121.07.

**Keywords:** Iterative Residual Rescaling, Term Frequency, Inverse Document Frequency, Latent Semantic Indexing, Pre-Processing.

## 1. Introduction

Mining of selected document is the key element of any study. Suppose you need some documents related to sentiment analysis, there are lot of PDF files available online or offline. You can download hundreds related file in large scope i.e. documents of sentiment analysis. But at the time of study, you need only those document which are related to phrase "sentiment analysis using supervised learning". Instead of finding related document manually, there should be a method which can automatically find those documents which are related to this text. LSI can provide a help for such issues. LSI is very easy to understand, implement and use. Results of LSI are very decent and faster compared to other methods. It aims to find the most representative features for document representation rather than the most discriminative ones [1] [2]. Major purpose of cosine similarity measure in LSI is to rank the data with respect to query, where data means stored documents and query is user requirement in text format. Here authors proposed and implemented an LSI (Latent Semantic Indexing) approach using preprocessing & without preprocessing and concluded that both algorithm will work correctly but algorithm with preprocessing have less processing time and greater manual work (finding stop words, how to remove them, how to convert in lower case etc.) and algorithm without preprocessing is the reverse of previous one. Purpose of both algorithms with pre-processing and without pre-processing is same. Both rank the documents in descending order with respect to query. Here we are supposing following table to represent such concept. In Table-1, D is document, A is Algorithm with Pre-Processing, and B is Algorithm without Pre-processing.

Table-1: Supposing Values of D Based on A and B.

| D | A | B | Ranking on A | Ranking on B |
|---|---|---|---|---|
| D1 | 0.8 | 0.7 | D2 | D2 |
| D2 | 0.9 | 0.8 | D1 | D1 |
| D3 | 0.4 | 0.2 | D4 | D4 |
| D4 | 0.6 | 0.5 | D3 | D3 |

In Table-1, Column-1 contains documents, Column-2 contains similarity cosine values using Algorithm-1, and Column-3 contains similarity cosine values using Algorithm-2. Resultant columns Column-4 and Column-5 representing the ranking of documents i.e. ranking of documents through both algorithms is same. Now, here we prove this supposition.

## 2. Related Material

In cluster tree [3], hybrid similarity has been measured by using LSI and LSI is used to cluster clinical document [25]. Authors in [4] have applied LSI to find representation of concept by mapping the terms and phrases with document and then clustering them. LSI (Latent Semantic Indexing) and ICA (Independent Component Analysis) [5] [6] have been used to find latent semantic structures in dataset each structure is a linear combination of the original features i.e. words. Using LSI approach, information retrieval methods has been proposed by the authors [7] using text documents. Sprinkling [9] which is the extension of LSI to supervised classification tasks and generating revised document representations that can be used by any technique founded on the vector space model. As LSI ignores class labels of training documents, sprinkling can handle such issue. Real world applications of topic modeling is limited due to issues of scalability. RLSI (Regularized Latent Sematic Indexing) is designed for parallelization and can handle large dataset without reduction of input vocabulary [11]. TF-IDF (term frequency-inverse document frequency), LSI (latent semantic indexing) and multi-word is used for extraction of feature which is helpful for identification of important words in a text document [12]. Main goal of Probabilistic Latent Semantic Analysis (PLSA) is to model co-occurrence information under a probabilistic framework in order to discover the underlying semantic structure of the data [13]. Multilevel Latent Semantic Association method grouped the words in aspect expression for aspect expression of latent topic structure [14]. General Text Parser (GTP) based on LSI, parse a huge collection of documents and create a vector space information retrieval model for subsequent concept-based query processing [15]. Sentiment analysis means analyzing the people opinion as positive or negative [17]. The research on sentiments and opinions appeared in 2001 [18] and 2002 [19]. LSI and Machine learning has been used for multi-lingual sentiment analysis [16]. To improve the efficiency of LSI, different researchers is working on different extensions of LSI i.e. SVR (Singular Value Rescaling) based on LSI made experiments on TREC dataset showing the 5.9% best results than LSI [20], dynamic hybrid cut improves the effectiveness of the LSI approach for detecting concerns in source code [21] and a term-to-concept projection matrix has been developed to reduce dimension for decreasing the bottleneck of LSI [24]. Extended method based on LSI is able to filter the unwanted emails of Chinese and English [23]. In advanced search, human not only require index term information but also concept and ideas. Such concept based searching and automatic key extraction can be done through LSI [26]. After the comparative study of multi-words, TF-IDF and LSI on text classification, the experimental results is showing that LSI has best performance than other two techniques [22]. LSI can resolve the problem of lexical matching by using statistically derived conceptual indices [10]. Authors [8] observed that after the evaluation of documents, LSI performed 40% better compared to exact term matching techniques.

## 3. Documents Ranking through LSI Algorithms

Latent Semantic Indexing (LSI) proposed by Deerwester in 1990 is an efficient information retrieval algorithm [7]. Basically in LSI, there is cosine similarity measure between coordinates of a document vector and coordinates of query vector. If this value is 1, means document is 100% closer to query, if it 0.5 means document is 50% closer and it is 0.9 means document is 90% closer with query.

Now the major point is that how we can find the coordinates of each document and query. Singular Value Decomposition (SVD) can determine the points or coordinates of documents and query. Through SVD, three matrices S, V and U can be determined by a matrix which will be used for further processing. To determine the values of such variables, SVD requires a matrix. Matrix consists of rows and columns containing integer values while here inputs are different text documents. Feature matrix can be obtained by calculating the frequencies of each word. It means, first of all we will make feature matrix from all documents and then will calculate SVD as shown in Algorithm in Table-2 from line 1-4. Line 5 and 6 will made a matrix for query. After this supporting variables S, V and U will be calculated by using numpy (Numeric Python). Now, from S, coordinates of all documents will be determined and these coordinates will be emerged with query to find query coordinates. At last, cosine similar function will be applied on these coordinates to fined closest documents to query.

### 3.1. Algorithm for Documents without Pre-Processing

We have checked above algorithm by taking three documents (d1="talcum powder has beautiful fragrance", d2=" talcum powder is white color", d3="black cat talcum powder") and a query (qry=" talcum powder is black cat") as input. In advance we know that d3 is very closest to query. Table-3 is depicting the results of given inputs.

Table-2: Algorithm of LSI without Preprocessing

| | |
|---|---|
| 1. | **Input:** All Documents and Query |
| 2. | Tokenize All Documents<br>**Token**=Token (All Documents) |
| 3. | Take the Union Set of Tokenized Documents<br>**UnionT**=Union (Token) |
| 4. | Make Frequency Matrix from UnionT<br>**fMat**=FrequencyMatrix (UnionT) |
| 5. | Make Query Matrix |
| 6. | qMat=QueryMatrix (Token (Query)) |
| 7. | Decompose Frequency Matrix in U,S,V using SVD from USVT |
| 8. | Determine V from VT |
| 9. | Find UK,Vk and SK |
| 10. | UK = Extracting first two column of U |
| 11. | VK = Extracting first two column of V |
| 12. | SK= Extracting first two column and row of S |
| 13. | Each row of V relates to Coordinates of Document |
| 14. | Find Coordinates of Query from **q = qTUkSk-1** |
| 15. | First we will find SK inverse from SK-→10 |
| 16. | Second q transpose from Query Matrix →4 |
| 17. | UK is already determined → 8 |
| 18. | Now, find **q = qTUkSk-1** |
| 19. | q have coordinates of query |
| 20. | Find dot product of q with each document coordinates (→13) |
| 21. | Sort dot product values in descending order |
| **22.** | Output Ranking of Documents with respect to query |

Table-3: Results of Algorithm-1

| |
|---|
| 1. d1="talcum powder has beautiful fragrance"<br>2. d2=" talcum powder is white color"<br>3. d3="black cat talcum powder"<br>qry=" talcum powder is black cat" |
| **Tokens**<br>['telcome', 'powder', 'has', 'beautiful', 'fragrence']<br>['telcome', 'powder', 'is', 'white', 'color']<br>['black', 'cat', 'telcome', 'powder'] |
| **Token Sets**<br>set (['beautiful', 'fragrence', 'has', 'telcome', 'powder'])<br>set (['color', 'is', 'white', 'telcome', 'powder'])<br>set (['telcome', 'black', 'powder', 'cat']) |
| **Union**<br>set (['beautiful', 'fragrence', 'color', 'is', 'cat', 'black', 'powder', 'white', 'has', 'telcome']) |

| |
|---|
| **Feature Matrix**<br>[ [1 1 0 0 0 0 1 0 1 1]]<br>[ [0 0 1 1 0 0 1 1 0 1]]<br>[ [0 0 0 0 1 1 1 0 0 1]] |
| **Query Matrix**<br>[ [0 0 0 1 1 1 1 0 0 1]] |
| **S**<br>[ 2.94984103 0. 0. ]<br>[ 0. 1.73205081 0. ]<br>[ 0. 0. 1.51605999] |

| |
|---|
| **V**<br>[ [-0.605 0.707 -0.364]]<br>[ [-0.605 -0.707 -0.364]]<br>[ [ -5.15 9.697 8.568]] |
| **U**<br>It is large matrix, we will display UK. |
| **SK**<br>[ [ 2.94984103 0. ]<br>[ 0. 1.73205081]] |
| **VK**<br>[ [-0.605 0.707]<br>[-0.605 -0707]<br>[-0.515 9.697]] |
| **UK**<br>[ [ -2.05405238e-01 4.08248290e-01]<br>[ -2.05405238e-01 4.08248290e-01]<br>[ -2.05405238e-01 -4.08248290e-01]<br>[ -2.05405238e-01 -4.08248290e-01]<br>[ -1.74754886e-01 6.56816799e-16]<br>[ -1.74754886e-01 6.56816799e-16]<br>[ -5.85565363e-01 2.46574729e-16]<br>[ -2.05405238e-01 -4.08248290e-01]<br>[ -2.05405238e-01 4.08248290e-01]<br>[ -5.85565363e-01 2.46574729e-16]] |
| **Coordinates of All Docs from VK**<br>[-0.605,0.707], [-0.605, -0.707], [-0.515, 9.697189] |
| **SK Inverse**<br>[ [0.33900, 0.0], [0.0, 0.57735]] |

| |
|---|
| **Coordinates of Resultant Query from Query Matrix**<br>**UK and SK-1**<br>[ [-0.58513178 -0.23570226]] |
| **Qry= telcome powder is black cat**<br>**Results**<br>D1= telcome powder has beautiful fragrence= 0.319826412535<br>D2= telcome powder is white color= 0.887280361339<br>D3= black cat telcome powder= 0.927572256443 |

| Document | Tokens | Order of Removing Stop Words | Out Put |
|---|---|---|---|
| Mining is a…. big field | Mining, is, a…, big, field | i) Remove stopw<br>ii) Remove exStopw | i) Mining, a…, big, field<br>**ii) Mining, a, big, field** |
| Mining is a…. big field | Mining, is, a…,big,field | i) Remove exStopw<br>ii) Remove stopw | i) Mining, is, a, big, field<br>**ii) Mining, big, field** |

From Table-3, it is clear that d3 (92%) is very close to query, d2 (88%) is close after d1 and d3 (31%) is close after d2.

**3.2: Validation of Algorithm-1:** To check the validity of algorithm, we can take a document similar to query. In above Algorithm1 when we have assigned another document d4 (d4=telcome powder is black cat) same as query (Qry= telcome powder is black cat) then result of d4 (100%) was 1.0 i.e. algorithm is working well, because query and d4 have same contents.

Table-4: Results of Algorithm-1 by taking input document same as Quer

| |
|---|
| Qry= telcome powder is black cat |
| Results |
| D1= telcome powder has beautiful fragrence= 0.319826412535 |
| D2= telcome powder is white color= 0.887280361339 |
| D3= black cat telcome powder= 0.927572256443 |
| D4= telcome powder is black cat= 1.0 |

**3.3: Algorithm for Documents with Pre-processing** In pre-processing there is lot of work i.e. converting each token in lower case, remove stop words, lemmatization, stemming, seeding etc. Here we are removing stop words and converting them into lower case. We have collected stop words (stopw) and special stop words (exStopw) from union result of algorithm-1 in Table-1. First we will remove all stop words from exStopw then from stopw, differ is shown in following Table-7.

Table-7: Steps for Stop Words Removal

```
stopw=["This","being", "as", "we", "have", "where", "been", "has", "had", "is","the",".","\n","on","in","of","from","to","I","we","it","there","for","their","our","and","due","a","this", "that","about","through","or","may","be","an","by","etc","can","also","these"]
exStopw=[".",",",":",":",";","?","/"]
```

From above Table-7, if we choose first process then final output will be “mining, a, big, field” which contains stop word ‘a’ because in input document there is a word ‘a…’ while second process have output “mining, big, field” which have removed all stop words. Now adding following code in Table-2 from line 3 to 5.

Table-8: Algorithm-2 LSI with Pre-Processing

```
Input: Different Documents and a Query
OutPut: Ranking of Documents Related to Query
stopw=["This","being", "as", "we", "have", "where", "been", "has", "had", "is","the",".","\n","on","in","of","from","to","I","we","it","there","for","their","our","and","due","a","this","that","about","through","or","may","be","an","by","etc","can","also","these"]
exStopw=[".",",",":",":",";","?","/"]
tokens = Union (tokenize (All Documents))
## Tasks
## (i) Count last characters related to exStopw
## (ii) Delete last counted characters from word
## (iii) Count last characters related to Stopw
## (iv) Delete last counted characters from word
Remove last character of word
For word in tokens
If word ends with any element of exStopw
        Do Task (i) and (ii)
If word ends with any element of Stopw
        Do Task (iii) and (iv)
```

Again we took similar inputs as in Algorithm-1 i.e. three documents (d1="talcum powder has beautiful fragrance", d2=" talcum powder is white color", d3="black cat talcum powder") and a query (qry=" talcum powder is black cat").-2. Following Table-9 is showing the results of Algorithm-2.

Table-9: Results of Algorithm2

| |
|---|
| Qry= telcome powder is black cat |
| Results |
| D1= telcome powder has beautiful fragrence= 0.422535 |
| D2= telcome powder is white color= 0.6180361339 |
| D3= black cat telcome powder= 0.9989216 |

From Table-9, it is clear that d3 (99%) is very close to query, d2 (61%) is close after d1 and then d3 (42%) is close after d2. In Table-10, A means Algorithm-1 and B means Algorithm-2.

Table-10: Similarity Percentages of documents from Both Algorithms

| D | A | B | Ranking on A | Ranking on B |
|---|---|---|---|---|
| D1 | 0.319826 | 0.422535 | D3 (31%) | D3 (42%) |
| D2 | 0.8872803 | 0.618036 | D2 (88%) | D2 (61%) |
| D3 | 0.9275722 | 0.998921 | D1 (92%) | D1 (99%) |

Hence from Table-10, it is clear that similarity percentages of documents with query is different in both algorithm but the results on base of descending order is same i.e. D3>D2>D1. This result is same as we have supposed in Table-1.

### 3.4. Comparison of Both Algorithms

Now we are taking abstract of 5 papers as (D2>D1>D3>D4>D5), manually we have checked, where D2 is very closely relevant to query while D5 is not related with query.

Table-11: Input Documents as abstract of different papers.

| | |
|---|---|
| D1 | This paper presents an unsupervised approach to aspect-based opinion polling from raw textual reviews without explicit ratings. The key contribution of this paper is three-fold. First, a multi aspect bootstrapping algorithm is proposed to learn from unlabeled data aspect-related terms of each aspect to be used for aspect identification. Second, an unsupervised segmentation model is proposed to address the challenge of identifying multiple single-aspect units in a multi-aspect sentence. Finally, an aspect based opinion polling algorithm is presented. Experiments on real Chinese restaurant reviews show that our opinion polling method can achieve 75.5% precision performance. |
| D2 | In this paper, we propose a review selection approach towards accurate estimation of feature ratings for services on participatory websites where users write textual reviews for these services. Our approach selects reviews that comprehensively talk about a feature of a service by using information distance of the reviews on the feature. The rating estimation of the feature for these selected reviews using machine learning techniques provides more accurate results than that for other reviews. The average of these estimated feature ratings also better represents an accurate overall rating for the feature of the service, which provides useful feedback for other users to choose their satisfactory services. |
| D3 | The “Aspect Based Sentiment Analysis” task focuses on the recognition of aspect term and category and classification of emotions (positive, negative, conflict, neutral) in restaurant reviews for the aspect. In this paper we propose the system for recognizing aspects and analyzing the sentiments using SVM for the restaurant review dataset. We compare the performance of the system with well-known KNN classifier. |
| D4 | Spam Detection Consumers increasingly rate, review and research products online (Jansen, 2010; Litvin et al., 2008). Consequently, websites containing consumer reviews are becoming targets of opinion spam. While recent work has focused primarily on manually identifiable instances of opinion spam, in this work we study deceptive opinion spam—fictitious opinions that have been deliberately written to sound authentic. |
| D5 | This research paper represents a multi-agent system, which have four Agents named as Knowledge Acquisition Agent, Attendance Agent, Decision Making Agent and Communication Agent that works together to that automatically gets inputs, manipulates the data, prepares timetable as well as keeps the record of students’ attendance and makes communication with its environment in an automatic fashion through sensors. All the agents work like human agents, which is one of the basic aims of computer technology. This work depicts an idea to integrate the Human Expertise, Information as well as the Biometric Technologies to solve real world problems. Feedback may be used as a learning element in the processing of the Multi-agent system. Snapshots (i.e., time table preparation, Attendance records, decision about absenteeism etc) depict how the various results are being provided by this multi-agent system to help human. This system can easily be implemented through adaptation of Biometric Technology and may also be used for employees’ attendance record as well as for security purposes, in future research. |
| Query | The rating estimation of the feature for these selected reviews using machine learning techniques and experiments on real Chinese restaurant reviews provides more accurate results than that for other reviews. |

After taking above documents and query as inputs for Algorithm-1 shown in Table-2 and Algorithm-2 shown in Table-8, we have obtained the following results from both as shown in Table-12. In Table-12, A means coordinates of Algorithm-1 and B means coordinates of Algorithm-2.

Table-12: Results of Both Algorithms

| D | A | B | Similarity Values based on A | Similarity Values based on B |
|---|---|---|---|---|
| D1 | [-0.26306995581813425, 0.17706615964420416] | [-0.26306995581813425, 0.17706615964420416] | 0.975432470662 | 0. 98282967271 |
| D2 | [-0.48315134771552765, 0.763917350444424] | [-0.48315134771552765, 0.763917350444424] | 0.980494811666 | 0. 999986475172 |
| D3 | [-0.3200168157115258, 0.1295143355108754] | [-0.3200168157115258, 0.1295143355108754] | 0. 929122463932 | 0. 898978659695 |
| D4 | [-0.09911747894964751, 0.053488425748337724] | [-0.09911747894964751, 0.053488425748337724] | 0. 964179178316 | 0. 891008231649 |
| D5 | [-0.7649339426218684, -0.604518719389029] | [-0.7649339426218684, -0.604518719389029] | 0. 13823609022 | 0. 0321949520955 |
| Query Coordinates | [-0.30767381 0.29438224] | [-0.30767381 0.29438224] | 1.0 | 1.0 |

Table-12 is representing the results of both algorithms on five documents. Manually we have selected D2 is very close to query and D5 is very far from query and ranking was D2>D1>D3>D4>D5. From results of both algorithms D2 (98% from Algorithme-1, 99% from Algorithm-2) is very close to query and D5 (13% from Algorithme-1, 32% from Algorithm-2) is very far from query. Hence from both algorithms ranking is D2>D1>D3>D4>D5. Now to obtain processing time, we have find size of each matrix as shown in Table-13.

Algorithm-2. In Table-13, A means Algorithm-1 and B means Algorthm-2.

Table-13: Size of Matrices in Algorithm-1 and

| | U | V | S | UK | Query Matrix | Feature Matrix |
|---|---|---|---|---|---|---|
| A | 86436 | 25 | 5 | 588 | 294 | 1470 |
| B | 53824 | 25 | 5 | 464 | 232 | 1160 |

Table-14: Different attempts on both Algorithms

| Samples | Algos | D1 | D2 | D3 | Automatic Identified Similar Doc with Query | Manually Identified Similar Doc with Query | Size of Feature Matrix | Size of U Matrix |
|---|---|---|---|---|---|---|---|---|
| Sample-1 | Algo-1 | 0.00731759340968 | 0.09664019845 | 0.99404749508 | D3 | D3 | 789 | 69169 |
| | Algo-2 | 0.202726610236 | 0.0277404903936 | 0.999587952509 | D3 | D3 | 648 | 46656 |
| Sample-2 | Algo-1 | 0.561377445816 | 0.853152864627 | 0.791933831548 | D2 | D2 | 840 | 78400 |
| | Algo-2 | 0.989564559362 | 0.998464409368 | -0.0567068171327 | D2 | D2 | 684 | 51984 |
| Sample-3 | Algo-1 | -0.0424976035434 | 0.780859494526 | 0.998660969105 | D3 | D3 | 780 | 67600 |
| | Algo-2 | 0.113754028079 | 0.741611395428 | 0.99218523483 | D3 | D3 | 609 | 41209 |
| Sample-4 | Algo-1 | -0.343125747237 | 0.939289476989 | 0.939289476989 | D2,D3 | D2,D3 | 780 | 67600 |
| | Algo-2 | 0.212232168853 | 0.977219272479 | 0.977219272479 | D2,D3 | D2,D3 | 609 | 41209 |
| Sample-5 | Algo-1 | 0.784547360818 | -0.617292208796 | -0.627806233247 | No | No | 426 | 20164 |
| | Algo-2 | 0 | 0 | 0 | No | No | 318 | 11236 |

Hence it is clear that U, UK, Query Matrix and Feature Matrix of Algorithm-2 have less size then that of Algorithm-1. But for Algorithm-2, we will consider some time for Pre-Processing.

**Conclusion**

Text based intelligent information processing is the requirements of each internet user. Then use search engine for retrieving information on the bases of sentence not on bases of particular word. Also users of computers want to search the documents from existing thousand stored documents. It is very hard to search out required documents from stored document manually. There is lot of work in related to such issue. After exploring the all techniques, LSI is a best method for retrieving the information. LSI has better semantic and statistically quality [22] and text retrieval is the current literature of LSI, we have implemented two algorithms (without Pre-Processing and with Pre-Processing) of LSI and found that both results are same with respect to ranking of documents. Here in Tabe-14 we have made different attempts on these algorithms to find out the maturity of these algorithms.

Here we took five samples of documents. Each sample consist of three (D1, D2, D3) text documents.

Sample-1: Manually we have considered D3 is very close to Query. After applying both algorithms on sample-1, we have obtained percentages of D3 from Algorithm-1 99% and also 99% from Algorithm-2.

Sample-2: Manually we have considered D2 is very close to Query. After applying both algorithms on sample-2, we have obtained percentages of D2 from Algorithm-1 85% and 99% from Algorithm-2.

Sample-3: Manually we have considered D3 is very close to Query. After applying both algorithms on sample-3, we have obtained percentages of D3 from Algorithm-1 99 and also 99% from Algorithm-2.

Sample-4: Manually we have considered D2 and D3 is very close to Query. After applying both algorithms on sample-4, we have obtained percentages of D2 & D3 from both algorithms 93% & 97%.

Table-14 is representing that both algorithms is working well while processing time of Algorithm-1 is greater than Algorithm-2 because matrices' size of Algorithm-1 is greater than Algoritharithm-2. Following figure is representing that Algorithm-1 has greater processing time than Algorithm-2 with respect to 5-samples of documents.

**References:**


1. Jian-Tao Sun, Zheng Chen, Hua-Jun Zeng, Yu-Chang Lu, Chun-Yi Shi, and Wei-Ying Ma, Supervised Latent Semantic Indexing for Document Categorization. ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining. 535–538, 2014.
2. Quora User, What are the advantages and disadvantages of Latent Semantic Analysis? Sep 13, 2013. Visit on Date: 12-05-2015. http://www.quora.com/What-are-the-advantages-and-disadvantages-of-Latent-Semantic-Analysis
3. M. Varshana Devi, Cluster Tree Based Hybrid Document Similarity Measure, International Journal of Innovative Research in Computer and Communication Engineering Vol.2, Special Issue 1, March 2014.
4. Clustering Documents using a Wikipedia-based Concept Representation, Clustering Documents using a Wikipedia-based Concept Representation, 2008.
5. Deerwester, S., Dumais, S. T., Furnas, G.W., Landauer, T., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6), pp. 391–407, (1990)
6. Kolenda, T., Hansen, L. K.: Independent Components in Text. In M. Girolami, editor, Advance in Independent Component Analysis, chapter 13. Springer-Verlag, pp.235–256 (2000)
7. Neelam Phadnis, Framework for Document Retrieval using Latent Semantic Indexing, International Journal of Computer Applications (0975 – 8887) Volume 94 – No.14, May 2014.
8. 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.
9. Sutanu Chakraborti, Supervised Latent Semantic Indexing Using Adaptive Sprinkling, IJCAI 07, 2006.
10. Barbara Rosario, Latent Semantic Indexing: An overview, INFOSYS 240 Spring 2000.
11. Regularized Latent Semantic Indexing, 2011 ACM 978-1-4503-0757-4/11/07. SIGIR'11, July 24–28, 2011, Beijing, China.
12. M. Govindarajan, Sentiment Analysis of Movie Reviews using Hybrid Method of Naive Bayes and Genetic Algorithm, International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970) Volume-3 Number-4 Issue-13 December-2013.
13. Dan Oneata, Probabilistic Latent Semantic Analysis, 2011.
14. Guo, Honglei, Huijia Zhu, Zhili Guo, Xiaoxun Zhang, and Zhong Su. Product feature categorization with multilevel latent semantic association. in Proceedings of ACM International Conference on Information and Knowledge Management. 2009.
15. MAHESH T R, TEXT MINING: ADVANCEMENTS, CHALLENGES AND FUTURE DIRECTIONS, International Journal of Reviews in Computing, ISSN: 2076-3328, © 2009-2010 IJRIC & LLS.
16. Brett Bader, Multilingual Sentiment Analysis Using Latent Semantic Indexing and Machine Learning, SENTIRE, December 11, 2011.

17. Bing Liu. Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.
18. Das, Sanjiv and Mike Chen. Yahoo! for Amazon: Extracting market sentiment from stock message boards. Proceedings of APFA-2001. 2001.
19. Morinaga, Satoshi, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. in Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). 2002.
20. Hua Yan, Augmenting the power of LSI in text retrieval: Singular value rescaling. Data & Knowledge Engineering 65 (2008) 108–125. ELSEVIER.
21. Pieter van der Spek. Applying a dynamic threshold to improve cluster detection of LSI, Science of Computer Programming 76 (2011) 1261–1274, ELSEVIER.
22. Wen Zhang, A comparative study of TF-IDF, LSI and multi-words for text classification. Expert Systems with Applications 38 (2011) 2758–2765. ELSEVIER.
23. QING YANG, SUPPORT VECTOR MACHINE FOR CUSTOMIZED EMAIL FILTERING BASED ON IMPROVING LATENT SEMANTIC INDEXING, Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.
24. Jingmin HAO, Improving Latent Semantic Indexing with Concepts Mapping Based on Domain Ontology, 978-1-4244-2780-2. 2008 IEEE.
25. Choonghyun Han, Effect of Latent Semantic Indexing for Clustering Clinical Documents. 978-0-7695-4147-1. 2010 IEEE.
26. Ravina Rodrigues, Concept Based Search Using LSI and Automatic Keyphrase Extraction. Third International Conference on Emerging Trends in Engineering and Technology. 2010 IEEE.

1/15/2021