International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-2, July 2020

# LSA Based Text Summarization

**Shrabanti Mandal, Girish Kumar Singh**

***Abstract*: *In this study we propose an automatic single document text summarization technique using Latent Semantic Analysis (LSA) and diversity constraint in combination. The proposed technique uses the query based sentence ranking. Here we are not considering the concept of IR (Information Retrieval) so we generate the query by using the TF-IDF(Term Frequency-Inverse Document Frequency). For producing the query vector, we identify the terms having the high IDF. We know that LSA utilizes the vectorial semantics to analyze the relationships between documents in a corpus or between sentences within a document and key terms they carry by producing a list of ideas interconnected to the documents and terms. LSA helps to represent the latent structure of documents. For selecting the sentences from the document Latent Semantic Indexing (LSI) is used. LSI helps to arrange the sentences with its score. Traditionally the highest score sentences have been chosen for summary but here we calculate the diversity between chosen sentences and produce the final summary as a good summary should have maximum level of diversity. The proposed technique is evaluated on OpinosisDataset1.0.***

***Keywords: Text summarization, LSA, SVD, LSI and diversity constraint.***

## I. INTRODUCTION

The volume of digital data is spreading dramatically with explosively increasing the information of World Wide Web. Information searching from this extensive sea of information effectively is very inadequate while using traditional Information Retrieval (IR) technique. For this searching purpose text search engines some time give the significant outcome by filtering and analyzing the list of admissible documents. The text search engine based on keyword-based is enough to overwhelm the users by presenting the millions of output. That's why we need an approach to retrieve quickly the highly admissible documents. The text summarizer can be represented as one of the solutions to the users. Both text search and text summarization are considered as crucial technologies and also accompaniment each other. The information searching becomes easier if summary is used to represent each document [1].

Mcdonald et al. said that the purpose of text summarization can be classified by their aim, focus and coverage[5]. The aim or intent mentions to the potential application of summary. The purpose of summary can be categorized into three classes like inductive, informative and evaluative [23].

Inductive summaries help both for identifying the core concept of the source document and also to determine the text's importance. As the size of informative summary is more than the inductive so it sometime is used as a replacement of original documents. The evaluative summary is used for expressing the view points of the composer on a certain concept. In both generic or query relevant summary, focus means the scope of summary.

The coverage means how much aspects have been covered into the summary instead of single document or multi document summarizer. Gong et al. [24] has stated that two types of text summaries are available; one is Query-relevant summaries and other is Generic summaries. In Query-relevant summary, only these sentences are selected for including into summary which are pertinent with the query in traditional IR approaches. In this case, the summary is generated based on the given query, so it can be assumed that the overall concept of the source document do not achieve. In compare with query relevant summary, generic summary is capable to represent the overall idea of the given document. A generic summary is said to be good if it covers maximum important aspects of the input text keeping in mind the minimum overlapping. To output the high quality summary, it is a big challenge to the generic summarizer as neither query nor concept is supplied as input to the summarization process. In this paper we are trying to present an approach for text summarization. This approach is a mixing concept of generic and query based summarization. As we have already discussed about the benefits of both approaches, so we here successfully use the mixing concept with Singular Value Decomposition (SVD) for producing the rich summary. Finally we evaluate our summary with similar existing approaches by calculating the ROUGE-1 and ROUGE-2 score and get outperform.

## II. RELATED WORK

Summarization is considered as strong Natural Language Processing (NLP) exercise which needs semantic analysis, discourse processing and inferential interpretation (grouping of the content using world knowledge)[1]. So researchers have been doing their best on the NLP assignment for producing an enhanced summary. The working with summarization has been introduced in the late fifties. So several approaches and techniques have been implemented varies from single document to multi document summarization and even extractive to abstractive summarization.

Manuscript received on May 25, 2020.
Revised Manuscript received on June 29, 2020.
Manuscript published on July 30, 2020.
* Correspondence Author
**Shrabanti Mandal**, Department of Computer Science & Applications, Dr. Harisingh Gour Central University, Sagar, MP ,India. Email: shrabmandal@gmail.com
**Girish Kumar Singh***, Department of Computer Science & Applications, Dr. Harisingh Gour Central University, Sagar, MP ,India. Email: gkrsingh@gmail.com

That's why we can say that the list of summarization approaches becomes longer over the years. Luhu[14] stated that extractive approach of summarization usually used to extract the significant concepts by computing the frequency of term from the source document. H. P

Edmundson [13] explained the extractive approach by using title word, cue phase, key method, position method –surface level approach and Daniel Jacob Gillick [4] worked on naïve- Bayes based classifier to classify the sentences i.e. one of the machine learning based approach. Eduard Hovy and Chin-Yew Lin [7] have proposed a statistical approach for extractive summarization by using decision tree based on the sentence position. Gerald Salton [10] focused on graph based approach based on term frequency-inverse document frequency to prepare automatic indexing used for text summarization. Abstractive and extractive approaches are not a same concept. Sometimes, abstractive approach applies extractive concept to produce the abstractive summary [18]. Knight and Marcus [18] trained a model for squeezing the linguistic parse tree of a sentence for generating the reduced but yet superiorly syntactic form that named as reduction approach. Daume et al.[12] has great contribution to compression approach under the heading of Rhetorical Structure Tree for selecting the most relevant squeezed and ignoring the rest ones by using decision tree , this approach is named as compressive summarization. Among the several techniques of text summarization, mostly are based on extractive technique i.e. it helps to select the significant sentences, because non extractive or abstractive approach demonstrates itself as a moreover daring task for defining the source document into a deep semantic way, for explaining the linguistic meaning into a conventional depiction and searching new ideas additionally to represent the text and produce new concise text which carries the almost same meaning of source text.

SUMMARIST is one of an extractive approach to generate the summary of five different languages. SUMMARIST uses the following 'equation': Summarization = Topic Identification + Interpretation + Generation [8]. The evaluation of generated summary is also a daring task as there is no ideal or standard summary for source even though various evaluation approaches have been introduced for document summarization in general. In general the summary generated by extractive approach tries to stand near to the input document by extracting the sentences from source in consequence restricting the partiality that might emerge in a summary [14]. Even though summary doesn't need to be reasonable, the critic can make a view of satisfaction of the source. CGI/CMU text summarizer applies the Maximal Marginal Relevance(MMR) technique for computing the applicability of each sentence to the query given by user and in addition to this it selects sentences and includes into the summary [16]. The highly admissible sentences to the user's query and in addition the sentences having maximum diversity among itself have been chosen for final summary. SRA International Inc. offers a Knowledge Management (KM) system based on morphological analysis, name tagging and co-reference resolution for extractive summarization [15]. The KM uses a technique based on machine learning for computing the excellent consolidation of these characteristics merging with statistical information from the text collection for determining the optimal sentences for adding into a summary. R. Barzilay et al. [20] has implemented an approach for text summarization by determining the lexical chains from the source. Buckley et al.[3]has developed the Cornell/Sabir system which utilizes the proficiency of both text retrieval and document ranking of SMART, text search engine for adequately determining relevant text from input document. B. Baldwin and T.S. Morton have proposed a query based extractive text summarizer which is able to extract all relevant sentences carrying the phrases of query [2]. A sentence is acknowledged to be relevant with the terms in query if it relates to same individual, organization and event etc. Yihong Gong et al. [24] has evaluated the system generated summary with the man made summary by computing the Recall(R), Precision (P) and F-Measure. This study shows that IR-based method outperforms on the average.

## III. LATENT SEMANTIC ANALYSIS

LSA combines the algebraic and statistical techniques which help to find out the hidden structure of words and or between words, sentences and documents. The LAS method reads the source document and represents it as a set of sentences. Then term-document matrix has been produced for further processing. The term-document matrix is decomposed by an algorithm known as Singular value Decomposition (SVD). The SVD performs decomposition for tracing and modeling the relationship between words and sentences and it also assists for cutting down the noise. Overall LSA algorithm performs in three steps consisting of term-sentence/document matrix, apply SVD to term-sentence/document matrix and sentence selection for summary [6].

### A. Term-sentence/document matrix

The term-sentence/document matrix is the starting point of LSA algorithm. The term-sentence matrix is produced based on the Vector Space Model (VSM). The VSM represents the bags of words in matrix representing rows by terms and column by sentences. The process of matrix generation is very complex as this process follows the pre-processing algorithm which consists of tokenization, stop word removal and streaming.

### B. Apply SVD to matrix

SVD is hypothesis based on linear algebra. This hypothesis helps to decompose a rectangular matrix called A into three matrices like an orthogonal matrix U, a diagonal matrix D and the transpose of an orthogonal matrix V. The objective of SVD is to utilize the dimensional matrix set of data point and converting it to a lower dimensional matrix. It also used to cut down the term-by sentence matrix. This method also discovers the latent data during noise removal of noise. The processing of SVD can be represented by equation (1)

$$A = UDV^T \quad (1)$$

In equation (1),the eigenvectors of $AA^T$ matrix are stored in the columns of left eigenvectors named U matrix. Here U matrix is representing the concept-by term relation. D matrix stores the singular values of A in diagonal elements and non-diagonal elements are set to zero. This matrix is actually tracing the concept-by-concept relation.

The columns of V matrix are stored by eigenvectors of $AA^T$ matrix. V matrix is known as right eigenvector which traces the concept-by-document relation. The transpose of V matrix is represented by $V^T$ .The dimension of LSA space is converted to k dimensional space by cutting down dimension. The dimension changing is performed by the value of k for cutting down the unimportant descriptions for enhancing the output of information retrieval so value of k whether large or small does not matter.

### C. Sentence selection process

After creating the term-sentence/document matrix and decomposing this matrix, sentence selection process starts. For selecting the sentences, sentence ranking is important. Sentences have been arranged and ranked according to the calculated score. The summarization algorithm uses the score or rank for including the sentences to the output or summary.

- **Different algorithm for sentence selection**

Gong and Liu [24] first time used LSA for the purpose of document summarization so they are considered as pioneers. For doing so they began by generating term-sentence/document matrix. To create the term –sentence matrix, the authors identified the sentence-wise unique terms that's why applying SVD to matrix has two different views. First view is related to transformation aspect. This view represents the mapping between m-dimensional space spanned and r-dimensional singular vector space. The m-dimensional space spanned is created by vector of weighted term-frequency and r-dimensional singular vector space has all of its axes linearly-independent. The second view is concerned about semantic point. This view applies the SVD for getting the latent semantic analysis of the source document that is represented by term-sentence/document matrix. Actually this matrix is considered as representation of base ideas or r-dimensional linearly-independent base vectors. The outputs of SVD are a singular value matrix and right singular vector $V^T$. Every column of $V^T$ represents the individual sentences in the singular vector space. From the matrix $V^T$ , $q^{th}$ right singular vector have been extracted that means high indexed sentences have been selected for including into summary. This process repeats until q number of sentences has been selected for summary.

There are few weakness of technique used in [24]. This technique is used to select the sentences and the concepts of these selected sentences are considered as the same as fairly ideas of source document. This is encountered as one of the significant drawbacks of this technique. The numbers of relevant sentences from each topic are equal to dimension and the larger the sentence the less important concept is picked. The approach proposed by Steinberger and Jezek in[17] chooses the sentences into the matrix having highest length using sentence vector by vector shown in equation (2).

$$L = \sqrt{\sum_{j=1}^{n} V_{ij} \times D} \qquad (2)$$

In equation (2), L is considered as length of the score, eigenvector of matrix $A^TA$ is stored in the columns of V matrix and D is diagonal matrix carry the singular values of matrix A.

Murray et al. [9] has proposed a technique allowing that multiple sentences extraction from highest significant ideas of source document and allowed to put in the top most rows of the matrix rather than selecting the only most important sentences from each concept. The conclusion regarding number of sentences would be selected from each topic is decided by using matrix. The value for each topic is set by calculating the ratio of relevance singular value and total of all singular values. The approach proposed by Murray et al. related to sentence selection claims that it can overcome the issues of the approach introduced by Gong & Liu. This proposed approach is able to select multiple such sentences even if sentence doesn't have large cell value in the row of the relevant topic.

The study [17] has introduced an approach which uses the Cross method to enhance the performance of approach proposed by Steinberger and Jezek. In the proposed approach, initial steps like creation of input matrix and calculation of SVD are followed as same way as others do and after that matrix is applied for choosing the sentences to compose the summary. The pre-processing step is performing in between SVD calculation and sentence selection for eliminating the words which have less important in the document and keeping rest of terms as they are. Each concept is represented as a row of matrix and mathematically mean score of the sentences is computed. Then identify this cell which value is below the mean score and this cell value is set to zero. Makbule Gulcin Ozsoy [19] has followed all the steps with little bit modification.

## IV. PROPOSED ALGORITHM

We have designed an automatic summarization technique based on both query based and LSA. We here use the benefits of both above mention methods for better performance. The data flow of proposed model has been depicted in figure 1.Figure 1 actually shows the overall tasks and decision taken by the proposed technique. The summarization technique in general follows the following steps

- **Step 1**

Pre-processing- Pre-processing is performed on corpus followed by segmentation, tokenization, stop word removal and stemming respectively so that further analysis can be performed easily [21][22].

**Sentence segmentation:** Under the heading of sentence segmentation the corpus is represented as separate individual sentences. Like input document is $D = \{s_1, s_2, \ldots, s_n\}$ , where $s_i$ means $i^{th}$ sentence and n denotes the total number of sentences in the document.

**Tokenization:** Tokenization process represents each document as unique terms wise. So for example $D = \{t_1, t_2, \ldots\ldots\ldots, t_m\}$ , where $t_k$ means all unique terms and k=1,2,3……,m.

**Stop word removal:** It is considered that the term used least in a document has more important than the word used repeatedly. But English language has predefined list of such words known as stop word which have less important. So for smoothen the summarization process extract these stop words from corpus.

**Stemming:** This process is applied to clip ends or truncate of words to a common base form.

- **Step 2**

Term –Document Matrix: After cleaning the source document our next task is prepare the Term-Document Matrix.

In this matrix each row is shown the unique terms and the each column is represented by the individual sentences. If a particular term presents in a sentence then this cell is set to number of times that occurs otherwise cell is set to zero. In the next step we apply SVD i.e. already explained in the previous section to the term-document matrix

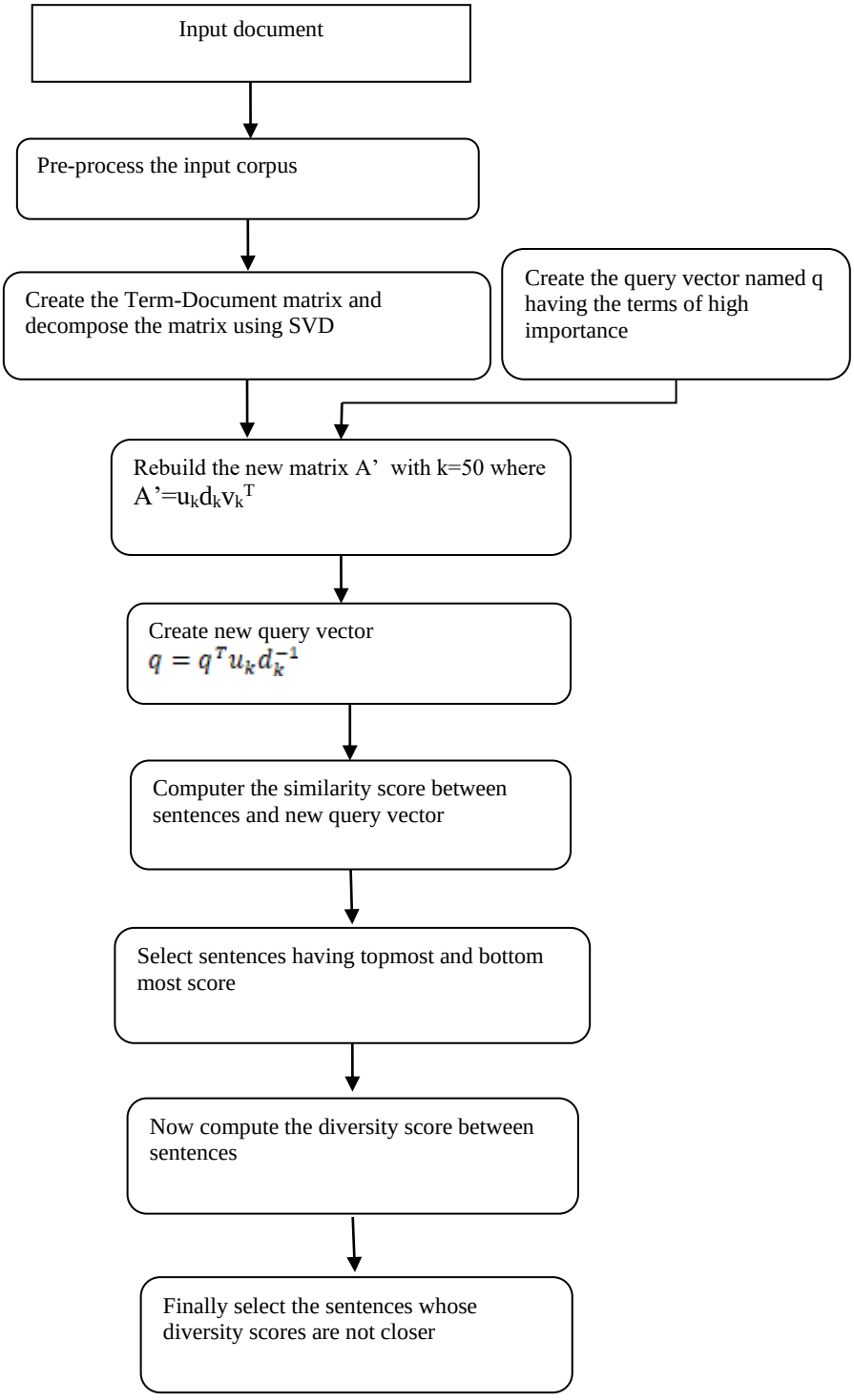


**Figure 1: Data flow for proposed model**

- **Step 3**

svd(A)= $udv^T$ where resultant matrix u is term vector that is linearly independent w.r.t. the relationship with the documents, matrix v is vector representation of documents whose components are linearly independent w.r.t. the relationships with terms in A. The d matrix contains the singular values arranged in descending order and these values show the relationships between other matrixes. The topmost values in d show the relevancy with considerable variance between terms and documents. The representation of u, v and d matrix are shown in figure 2,3 and 4 respectively.

```
> A.svd$u[1:10 ,1:5]
             [,1]         [,2]         [,3]         [,4]         [,5]
 [1,] -0.042924697 -0.060847457 -0.112653056  0.02383190  0.035810929
 [2,] -0.054135679  0.016241237 -0.126434091  0.10026590  0.043202487
 [3,] -0.171416467  0.366568329 -0.214278301  0.49813211  0.035583683
 [4,] -0.072282851  0.016273331 -0.156866836  0.08768759  0.045496852
 [5,] -0.025117393  0.004880168 -0.047661905  0.03214342 -0.004512544
 [6,] -0.008849491  0.009447094 -0.043929534  0.03522202 -0.003346815
 [7,] -0.390936920 -0.628229577 -0.429823014 -0.06556215  0.063831643
 [8,] -0.060121128  0.013643337 -0.002200886 -0.06390177 -0.186641473
 [9,] -0.439217582  0.333955023 -0.271889939 -0.11783437 -0.152924671
[10,] -0.009863997 -0.006293414 -0.008435150 -0.01207902  0.005620042
```

**Fig. 2: Representation of u matrix**

```
> A.svd$v[1:10,1:10]
             [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
 [1,] -0.11274562  0.06436264 -0.26677058  0.19029543 -0.01715341  0.11194350
 [2,] -0.12567078 -0.04287676 -0.05122408 -0.06525983  0.02880436 -0.07212621
 [3,] -0.09921132 -0.08845844 -0.04066970 -0.01835791  0.08343762  0.01325894
 [4,] -0.12409424  0.16489324 -0.05241528  0.11170359 -0.01675037 -0.04726303
 [5,] -0.06907423  0.05803707  0.03683035  0.08061506  0.05105557  0.03633624
 [6,] -0.10421293  0.05951201  0.05478727 -0.07865880  0.04141821 -0.10043680
 [7,] -0.16723405 -0.14089259 -0.17977774 -0.04662140  0.06051436 -0.03265835
 [8,] -0.13279656 -0.02535367 -0.03110535 -0.07972539 -0.02656558  0.06337062
 [9,] -0.06945987  0.06607813  0.04708592  0.08598521  0.05188852  0.03494794
[10,] -0.15631447  0.07005039  0.12556234 -0.08387226  0.07114325 -0.05650690
```

**Fig. 3: Representation of v matrix**

```
> A.svd$d
 [1] 12.7403504  6.8129560  6.0726931  5.4027401  5.1252925  4.8405236
 [7]  4.6642305  4.6154291  4.5443227  4.2388384  4.1446947  4.0003836
[13]  3.8899303  3.7193219  3.6727849  3.6325215  3.6047833  3.4951167
[19]  3.4168892  3.4032113  3.3757630  3.2877140  3.2671697  3.1730456
[25]  3.0442723  3.0162915  2.9900072  2.9596416  2.8652442  2.8455741
[31]  2.8047874  2.7717385  2.7411103  2.6684970  2.6493804  2.6154572
[37]  2.5848496  2.5358521  2.5042823  2.4489408  2.4210579  2.3623606
[43]  2.3422618  2.3185311  2.2585984  2.2274292  2.2133862  2.2100954
[49]  2.1823907  2.1485857  2.1220760  2.0742801  2.0539256  2.0283630
[55]  1.9947825  1.9446480  1.8897262  1.8787734  1.8337457  1.7862004
[61]  1.7313265  1.6819792  1.6620788  1.6307531  1.6254181  1.5792703
[67]  1.5467913  1.5062544  1.4704941  1.4373029  1.4283523  1.3682518
[73]  1.3316139  1.2825015  1.2432269  1.1974641  1.1618438  1.0965213
[79]  1.0617420  1.0247684  0.9825305  0.9308774  0.8206843  0.7824382
[85]  0.6659279  0.6324606  0.5459876  0.3555963
```

**Fig.4: Representation of d matrix**

- Step 4

Query Vector creation: Now we effort to generate the query vector. We here don't allow users to enter the query based on which summary will be generated. But we create the vector having with high score terms. The terms and query vector is shown in figure 5 and figure 6 respectively. The figure 5 shows the bag of words with its frequency in decreasing order. In query vector first column represents the terms in descending order and second column shows the weather term is considerable or not during summarization.

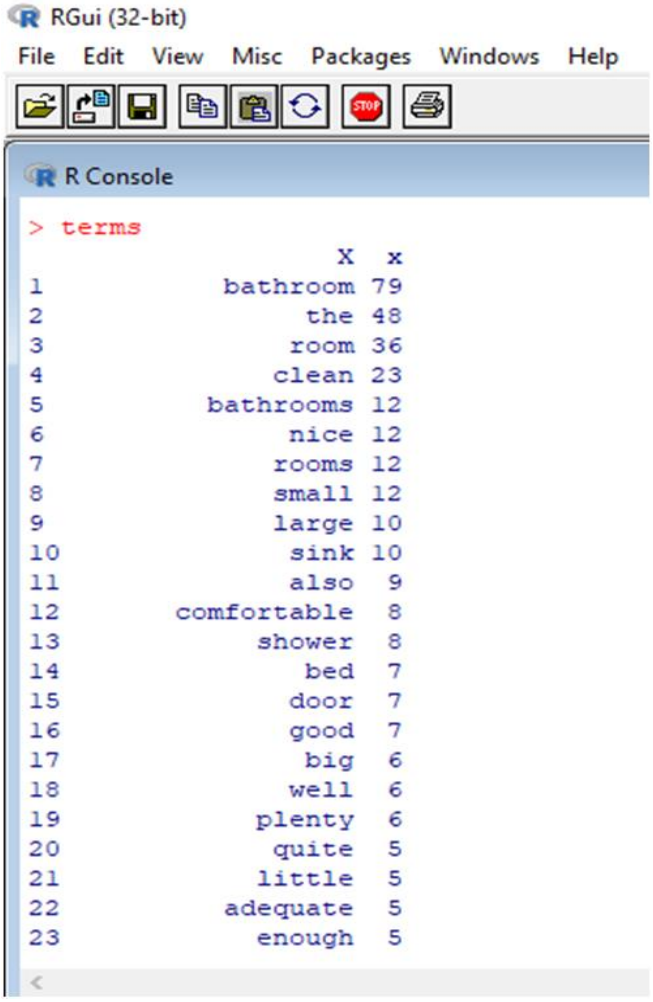


**Fig. 5: Representation of terms with frequency**

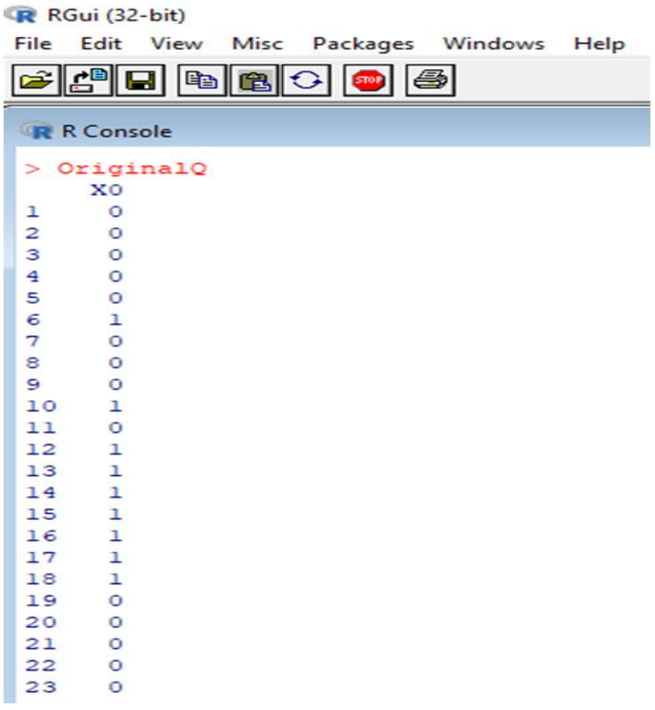


**Fig. 6: Query vector**

- **Step 5**

For avoiding the complication after decomposing the source matrix A can be recomposed by the product of the resultant three matrixes. To rebuild the new matrix A', only choose first k elements of the matrixes i.e. A' can be defined as $A'=u_k d_k v_k^T$. Here we have set the value of k=50. It is considered that the buzz generated by extraneous relations is erased. So the cell values of A' matrix disclose the latent relationship between terms and documents i.e. intellectual relationship [11].

- Step 6

Now for k=50, generate the new query vector by using the equation (3) and result shown in figure 7. The terms of equation (3) are already explained earlier.

$$q = q^T u_k d_k^{-1} \qquad (3)$$

```
> q=read.csv("D:\\q.csv", header=T)
> q
 [1] X.0.34297662  X.0.245921418 X0.331736905  X0.362470789  X.0.567175475
 [6] X0.255955633  X.0.126953931 X0.67188675   X.0.113892516 X0.132175343
[11] X.0.114967587 X.0.331385719 X0.256585965  X.0.278140359 X0.220356031
[16] X.0.393003572 X0.592829142  X0.5193695    X.0.170913767 X.0.171679377
[21] X0.348014193  X.0.211132735 X0.067357612  X0.646670191  X.0.16085989
[26] X0.852252499  X.0.175474453 X0.466340123  X0.561873781  X.0.314933156
[31] X0.492155941  X.0.07590161  X0.432235791  X0.302317757  X0.474113665
[36] X.0.555163304 X0.216173849  X.0.278567718 X0.281130095  X0.184378847
[41] X0.516187026  X.0.008561998 X.0.580516777 X.0.403063077 X.0.67823406
[46] X.0.063336934 X0.591219823  X0.240064027  X0.050018542  X.0.622379644
<0 rows> (or 0-length row.names)
```

**Fig. 7: Representation of new query vector**

- Step 7

Now compute the cosine similarity using the equation (4) and rank the sentences with score in descending order shown in figure 8.

$$sim(Q,D) = \frac{Q \times D}{|Q|X|D|} \qquad (4)$$

| SENTENCE NAME | Sqrt(Q) | Sqrt(V) | Q*D | SIMILARITY SCORE |
|---|---|---|---|---|
| 1 | 2.789480658 | 0.769105067 | -0.053668386 | -0.025015518 |
| 2 | 2.789480658 | 0.995562309 | 0.316132526 | 0.113721183 |
| 3 | 2.789480658 | 0.402910685 | 0.143150783 | 0.127368369 |
| 4 | 2.789480658 | 0.557722468 | -0.61929882 | -0.398069365 |
| 5 | 2.789480658 | 0.282631212 | -0.219765426 | -0.278750653 |
| 6 | 2.789480658 | 0.377475344 | 0.116553467 | 0.11069124 |
| 7 | 2.789480658 | 0.87351118 | -0.277385878 | -0.11383938 |
| 8 | 2.789480658 | 0.710880243 | -0.014109082 | -0.007115067 |
| 9 | 2.789480658 | 0.495009596 | -0.160711285 | -0.116153671 |
| 10 | 2.789480658 | 0.722071553 | -0.249815256 | -0.12402675 |
| 11 | 2.789480658 | 0.415830625 | 0.042873522 | 0.036961479 |
| 12 | 2.789480658 | 0.423583073 | 0.010988787 | 0.009300104 |
| 13 | 2.789480658 | 0.465642039 | -0.478554986 | -0.368431121 |
| 14 | 2.789480658 | 0.825802405 | 0.311851878 | 0.135214503 |
| 15 | 2.789480658 | 0.93989753 | 0.295285148 | 0.106937003 |
| 16 | 2.789480658 | 0.981344098 | 0.352445443 | 0.128749997 |
| 17 | 2.789480658 | 0.389539833 | -0.185619746 | -0.170824022 |
| 18 | 2.789480658 | 0.979330623 | 0.097271023 | 0.035606623 |
| 19 | 2.789480658 | 0.983472574 | 0.137258199 | 0.050032556 |
| 20 | 2.789480658 | 0.355284663 | -0.099784911 | -0.100685065 |
| 21 | 2.789480658 | 0.532159625 | -0.267721004 | -0.180350424 |
| 22 | 2.789480658 | 0.435230122 | -0.172431342 | -0.142027979 |
| 23 | 2.789480658 | 0.979833049 | 0.240786747 | 0.088096193 |
| 24 | 2.789480658 | 0.93918323 | 0.156778633 | 0.056818109 |
| 25 | 2.789480658 | 0.75080891 | 0.104765371 | 0.050022452 |
| 26 | 2.789480658 | 0.979506395 | 0.445223064 | 0.162947245 |
| 27 | 2.789480658 | 0.50360675 | -0.362732502 | -0.258209117 |

**Fig. 8: Similarity score**

- **Step 8**

Choose the both sentences having top most and bottom most similarity scores from the previous step. For including the sentences to the final summary, compute the similarity score between sentences to achieve the high level of diversity as we know that less similarity score achieves the higher level of diversity. Diversity is calculated by the equation (5)

$$div(s_j, s_i) = 1 - sim(s_j, s_i) \qquad (5)$$

Where $s_j$ and $s_i$ are sentences and i≠j.

## V. RESULT EVALUATION

We perform the experiment on the OpinosisDataset1.0 consisting 51 topics. We use almost 40 documents after cleaning –up the documents. This dataset provides the minimum 4 sets of abstractive summaries for each document. But our proposed technique works on the extractive basic. Therefore we have collected the several man made summaries from experts of summary maker. For evaluating the proposed technique, we compute F-measure score using the generic ROUGE-1 and ROUGE-2 score where ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. The approach based on LSA[24] presented the result of ROUGE-L score as recall, precision and f-measure are 0.6517,0.1715 and 0.2619 respectively. Our approach has noted the values of recall, precision and f-measure are 0.6698, 0.1972 and 0.38312 respectively. If we note the result of Steinberger and Jerek [17] approach i.e. recall, precision and f-measure then we get the values as 0.6473, 0.4708 and 0.5374 respectively. The comparison shown as tabular form in table 1 and graphical form in figure 9.

**Table 1: Average ROUGE-L score**

| Approach | Recall | Precision | F-measure |
|---|---|---|---|
| Gong and Liu | 0.6517 | 0.1715 | 0.2619 |
| Steinberger and Jerek | 0.6473 | 0.4708 | 0.5374 |
| Proposed | 0.6698 | 0.1972 | 0.38312 |

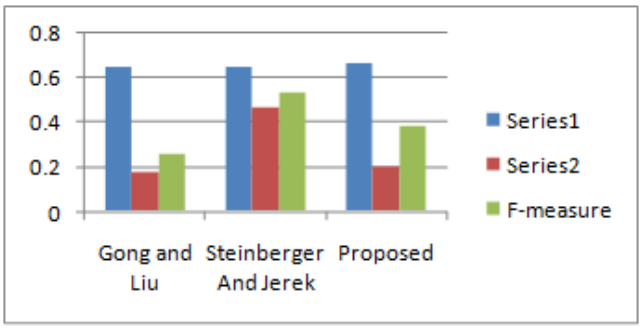


**Fig. 9: Average recall, precision and f-measure**

## VI. CONCLUSION

We have designed a LSA based text summarization technique with association with the diversity constraint. We evaluate our approach by computing the recall, precision and f-measure using the ROUGE score. Some approaches based on LSA like Gong et al. and Strinberger et al. are taken for comparing our results. The open source data sets named OpinosisDataset1.0 consisting 51 topics has been used for evaluating purpose.

Our approach outperforms the result of approach proposed by Gong et al.

We get the better recall score than the approach introduced by Strinberger et al. In future we can enhance the performance of our approach and may find the applicability for multi-document summarization.

## REFERENCES


1. A. Bellaachia, A. Mahajan, "Text Summary Using Latent Semantic Indexing and Information Retrieval Technique: Comparison of Four Strategies", In EGC 2004, vol. RNTI-E-2, pp.453-464.
2. B. Baldwin and T.S. Morton. Dynamic coreference based summarization. In Proceedings of The Third Conference on Empirical Methods in Natural Language Processing (EMNLP3), Granada, Spain, June 1998.
3. C. Buckley and et al.. The smart/empire tipster ir system. In Proceedings of TIPSTER Phase III Workshop. 1999.
4. D. J. Gillick "The Elements of Automatic Summarization" Electrical Engineering and Computer Sciences, University of California, May 2011.
5. D. McDonald et al., "Using Sentence-Sentence Heuristics to Rank Text Segments in TXTRACTOR, " MIS Dept., U. of Arizona, Tucson, AZ.
6. D. Oluwajana," Single-Document summarization using Latent Semantic Analysis", DOI: 10.13140/RG.2.1.4075.6320.
7. E. Hovy and C-Y. Lin, "automated text summarization and the SUMMARIST system" ,Information Sciences Institute of the University of Southern California, 1998.
8. E.Hovy and C. Lin. Automated text summarization in summarist. In Proceedings of the TIPSTER Workshop, Baltimore, MD, 1998.
9. G. Murray, S. Renals, J. Carletta "Extractive Summarization of Meeting Recordings", Centre for Speech Technology Research, University of Edinburgh, Scotland, 2005.
10. G. Salton and C. Buckley" Term Weighting Approaches in automatic Text Retrieval", Department of Computer Science, Cornell University, New York, November 1987.
11. G.W. Furnas, S.C. Deerwester, S.T , Dumais, T.K. Landauer, R.A. Harshman, L.A.Streeter, K.E. Lochbaum, Information retrieval using a singular value decomposition model of latent semantic structure. SIGIR Forum 51(2), 90–105 (2017).
12. H. Daume III and D. Marcu "A Tree-Position Kernel for Document Compression Proceedings of the Document Understanding Conference", Boston, MA. May 6-7, 2004.
13. H. P. Edmundson "New Methods in Automatic Extracting" Journal of the Association for Computing Machinery, Vol. 16, No. 2, April 1969.
14. H.P. Luhn, The Automatic Creation of Literature Abstracts. in Maybury, M.T. ed. Advances in Automatic Text Summarization. The MIT Press, Cambridge, 1958, 15-22.
15. http://www.SRA.com.
16. J. Goldstain, M. Kantrowitz, V. Mittal and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In Proceedings of ACM SIGIR ' 99, Berkeley, CA, Aug 1999.
17. Jezek and Steinberger "Automatic Text Summarization" The State of Art 2008 and the challenges, Bratislava. 2008.
18. K. Knight and M. Daniel "Statistical-Based Summarization One step: Sentence compression", Information sciences Institute and Department of Computer Sciences University of Southern California, 2002.
19. M. Gülçin Özsoy "Text Summarization Using Latent Semantic Analysis" Graduate School of Natural and Applied Sciences, Middle East Technical University, Ankara.2011.
20. R. Barzilay and M. Elhadad. Using lexical chains for text summarization", in Proceedings of the Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, Aug. 1997.
21. S. Mandal,G. K. Singh,A. Pal," PSO Based Text Summarization Approach Using Sentiment Analysis", Advances in Intelligent Systems and Computing ,Springer ,Vol 810,p.p.- 845-854, 2019, https://doi.org/10.1007/978-981-13-1513-8_86.
22. S. Mandal,G. K. Singh, A. Pal," Text Summarization Technique by Sentiment Analysis and Cuckoo Search Algorithm",Advances in Intelligent Systems and Computing, Springer , Vol 1025 ,p.p.- 357-366, 2020.
23. T. Firmin, and M.J. Chrzanowski, An Evaluation of Automatic ,1999.
24. Y. Gong and X. Liu. Generic Text Summarization Using Relevance Measure and latent Semantic Analysis. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 19 – 25, 2001.


## AUTHORS PROFILE

**Shrabanti Mandal**, obtained her Ph.D. in the field of Artificial Intelligence from National Institute of Technology, Durgapur, India in the year 2020. Currently she is working as Assistant Professor in Computer Science and Applications department at Dr. Harisingh Gour Central University, Sagar, Madhya Pradesh, India. Her areas of specialization include Fuzzy Logic, Data Mining and Soft Computing. She has published around ten research articles in International and National Journals and Conferences.

**Girish Kumar Singh** is Assistant Professor in the Department of Computer Science & Applications, Dr.Harisingh Gour Central University, Sagar (M.P). He completed his PhD from Jawaharlal Nehru University, New Delhi. He has more than thirty papers published in international journals and conferences. He has more than twelve years of teaching experience. His research areas include Data Mining, Soft Computing, and Distributed Computing.