PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

MAY 2020: IN SEMESTER ASSESSMENT (ISA) B.TECH. IV SEMESTER

UE18MA251- LINEAR ALGEBRA

MINI PROJECT REPORT

ON

Topic Modelling and Latent Semantics Analysis Using Singular Vector Decomposition

Submitted by

1.	Raghav Aggarwal	SRN PES1201800312
2.	Sidharth Pathak	SRN PES1201800142
3.	Rajdeep Sengupta	SRN PES1201800144

Branch & Section : Department of Computer Science, 4C

PROJECT EVALUATION

(For Official Use Only)

Sr.No.	Parameter	Max Marks	Marks Awarded
1	Background & Framing of the problem	4	
2	Approach and Solution	4	
3	References	4	
4	Clarity of the concepts & Creativity	4	
5	Choice of examples and understanding of the topic	4	
6	Presentation of the work	5	
	Total	25	

Name of the Course Instructor : P RAMA DEVI

Signature of Course Instructor :

Table of Contents

Table of Figures

List of Tables

1. Introduction

1.1 Problem Statement

Given a collection of text documents and find all the topics, inside the document, which are related, what are their keywords and what are the categories in which those topics can be labelled.

1.2 Background

Ever wondered how google or any other search engine gives all the related and relevant topics to the keyword which we type in the search bar, even if it is a single word? How does Google know that this keyword is associated with some other word?

Query information can provide interesting and helpful business insights. Knowing how often people search for products and brands can tell us how many people are interested in those products, as well as what people might associate with those brands. For example, clustering products into categories such as “beauty products” can give us insights into what people are searching for in popular trends. One of the difficulties that you could run across in trying to learn from topics that appear like this is that they show up in shortened contexts, such as tweets, reviews and queries, which means that you won’t learn from other terms that appear near them or with them. For instance, the word “Lipton” might show up frequently near words that are related to “tea” and to “soup”.

Word Co-occurrence is when the same terms or phrases appear frequently (not counting stop words) in documents that might rank highly for a query term, meaning that those words are likely semantically related to the terms they rank highly for.

Large amounts of raw data are collected every day and data is the new currency for major multinational companies. As more and more information is available, it becomes difficult to assess what we are looking for. If there are large paragraphs without having heading, then to determine which topic it belongs to, we would have to go through that paragraph or even a whole document to understand what it refers to. Let’s take an example, if there is an exam tomorrow and we have to look for a particular topic from our notes which are uncategorized. What if there is no heading in the newspaper, just large paragraphs are there, then it will be difficult to grasp that thing. It would take a lot of time to search for the relevant information

from that ocean of words which a person is looking for. So, we need tools and techniques to organize, search and understand vast quantities of information.

1.3 Aim

To cluster the topics present in various documents into categories in order to relate them with each other. It would save a lot of time instead of reading the whole paragraph or document to gain meaningful insights. ***Topic Modeling*** is the technique with the help of which we can perform this task.

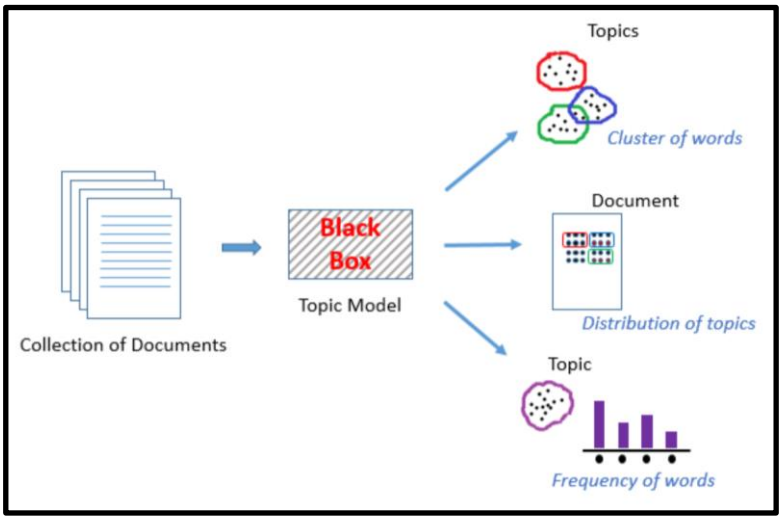

Figure 1:- **Need for Topic Modeling**

1.4 Scope

Topic modeling is a form of text mining, a way of identifying patterns in a corpus (bag of words). You take your corpus and run it through a tool which groups words across the corpus into 'topics'. Miriam Posner has described topic modeling as "a method for finding and tracing clusters of words (called "topics" in shorthand) in large bodies of texts."

It provides us with methods to organize, understand and summarize large collections of textual information by:

1. Discovering hidden topical patterns that are present across the collection of documents.
2. Annotating documents according to the topics.
3. Using these annotations to organize, search and summarize texts.

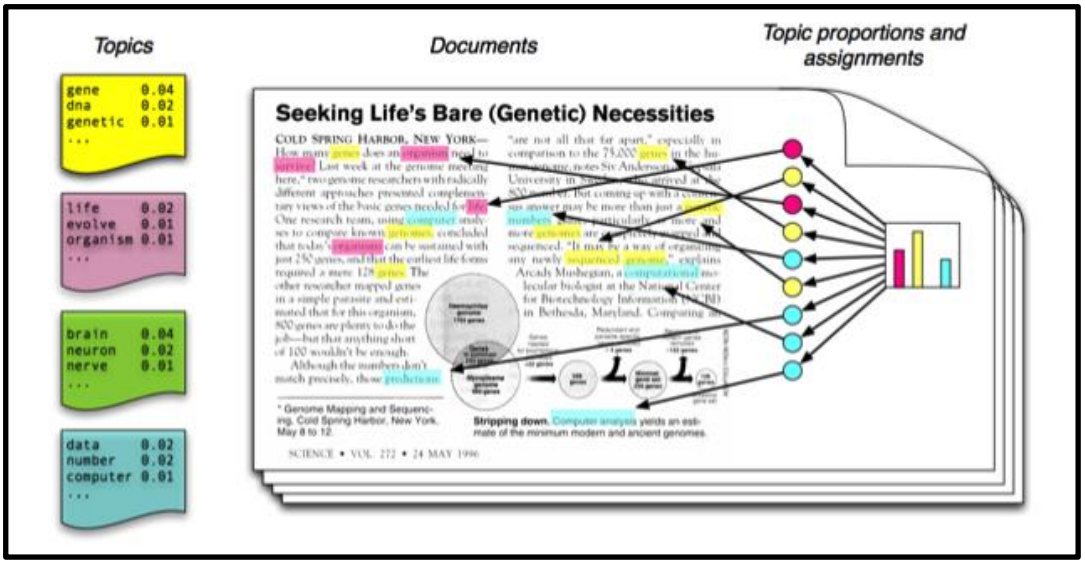

Figure 2:- Illustration from Blei D 2012- "Probabilistic Topic Models"

This image represents how the document can be labelled or categorized into topics. This description is inspired by the following illustration from David Blei's Article. In a good topic model, the words in the topic make sense, for example "navy, ship, captain" and "tobacco, farm, crops.". Here navy, ships are correlated so they come under the same topic.

1.5 How Linear Algebra Makes It Possible

There are various concepts and ways to implement Topic Modeling, but we mainly focus on the ***LATENT SEMANTIC ANALYSIS (LSA).*** As the name suggests is the analysis of latent i.e. hidden semantics in a corpora (plural for "corpus") of text. A collection of documents can be represented as a huge ***term-document matrix*** and various things such as how close two documents are, how close a document is to a user issued query, etc. can be inferred by cosine similarity. LSA uses ***Singular Vector decomposition (SVD)*** for ***dimensionality reduction*** to determine under which category the document falls into.

In English, a word has a lot of synonyms and each word can yield a topic or category and having different words having the same meaning can be considered as ***Noise*** in the text data. There are different words with multiple meanings, and all sorts of ambiguities that obscure the concepts to the point where even people can have a hard time understanding. present in the data. In order to reduce the noise and get the meaningful data the document-term matrix must be truncated.

LSA arose from the problem of how to find relevant documents from search words. The fundamental difficulty arises when we compare *words* to find relevant documents, because

what we really want to do is compare the *meanings or concepts behind the words*. LSA attempts to solve this problem by mapping both words and documents into a "concept" space and doing the comparison in this space.

Google uses LSA to assess the meaning of the written content on your blog or website.

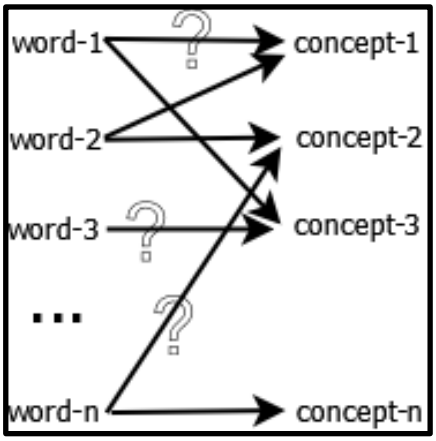

Figure 3:- Illustration of Noise in Data

2. Literature Review

2.1 Document-Term Matrix

LSA can use a term-document matrix which describes the occurrences of terms in documents; it is a sparse matrix whose rows correspond to terms and whose columns correspond to documents. This matrix is also common to standard semantic models, though it is not necessarily explicitly expressed as a matrix, since the mathematical properties of matrices are not always used.

Let us have a text collection composed of n documents containing m distinct terms. Documentd_j is represented by its document vector,

$$x_j = (x_{1j}, \ldots, x_{ij}, \ldots x_{mj})^T$$

where $x_{ij}$is a weight associated with the *i*-th word in the *j*-th document. The weight x_{ij} could be the number of occurrences of the term i in the document j or 0/1 depending if term *i* is present/absent in document *j* (binary weighting). A typical example of the weighting of the elements of the matrix is **tf-idf** (term frequency–inverse document frequency) that is the weight of an element of the matrix is proportional to the number of times the terms appear in each document, where rare terms are upweighted to reflect their relative importance.

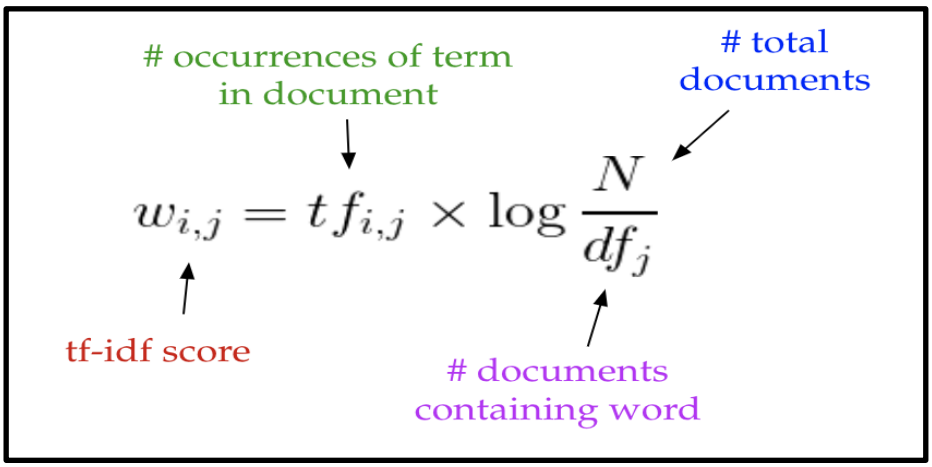

Figure 4:- **Formula to calculate tf-idf**

Let X be a $m \times n$ term/document matrix composed of document vectors x_j as columns,

$$\mathrm{X} = \begin{bmatrix} \mathbf{x}_1 & \dots & \mathbf{x}_j & \dots & \mathbf{x}_n \end{bmatrix} = \begin{bmatrix} \mathbf{t}_1{}^\top \\ \vdots \\ \mathbf{t}_i{}^\top \\ \vdots \\ \mathbf{t}_m{}^\top \end{bmatrix} = \begin{bmatrix} x_{11} & \dots & x_{1j} & \dots & x_{1n} \\ \vdots & & \vdots & & \vdots \\ x_{i1} & \dots & x_{ij} & \dots & x_{in} \\ \vdots & & \vdots & & \vdots \\ x_{k1} & \dots & x_{kj} & \dots & x_{kn} \end{bmatrix}.$$

Figure 5:- **Document-term Matrix**

Rows of *X* are 'term' vectors t_i and in the binary weighting scheme represent in which documents the term *i* is present. Columns of *X* are document vectors x_j.

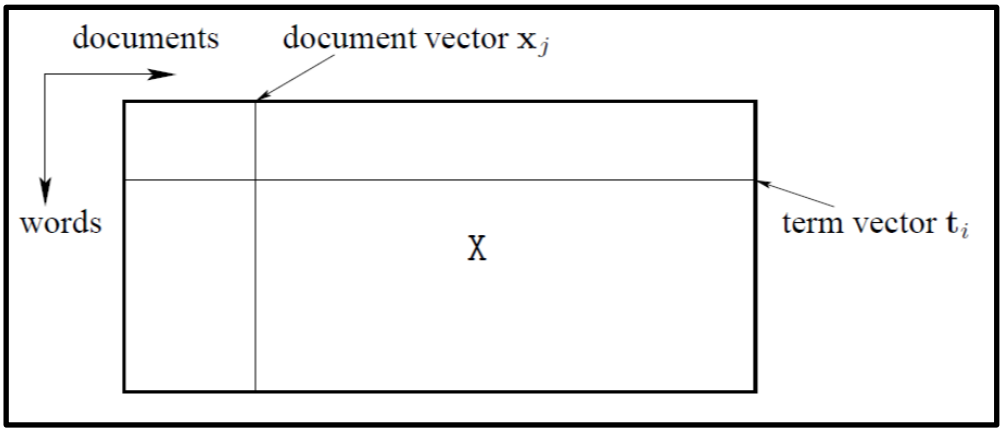

Figure 6:- **Schematic drawing of the term document matrix**

The document and/or term vectors can be normalized in various ways. A common practice is to normalize document vectors to have unit Euclidean (L2) norm. In that case the dot product between document vectors corresponds to the **cosine of the angle between the vectors**.

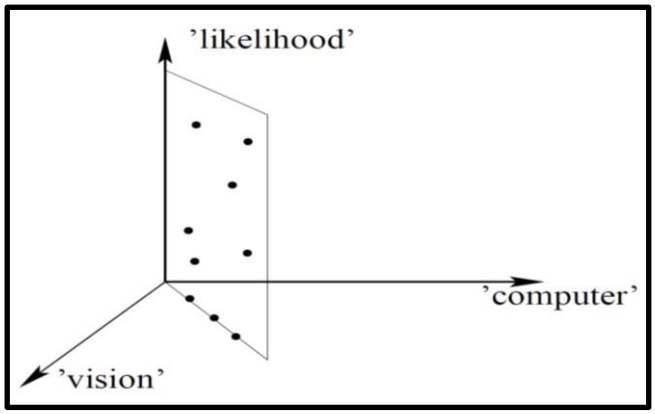

Figure 7:- **Documents in words space**

This example shows documents in the three-dimensional space of three terms. Coordinates of a document represent the number of occurrences the term appears in the document. All the shown documents have an equal number of occurrences of term ’computer’ and term ’vision’. This might indicate that these two terms appear always together. But we cannot say for example, whether the two terms appear close together or in what order, since we are using only the ’bag of words’ model. The term ’likelihood’ has a varying number of occurrences in different documents. There are three documents (the three points in the ’computer-vision’ plane) which do not contain the term ’likelihood’ at all.

Documents as ’points’ in the word space. In the term/document representation, documents could be thought of as points in the term space. Dimensions of the term space correspond to various terms and coordinates of the document are determined by e.g. the number of occurrences of the word in the document, see figure 6.

Let D be a $n \times n$ document similarity matrix,

$$D = X^T X$$

In the binary weighting model, element $d_{ij} = x_i^T x_j$ represents several distinct words document *i* and document *j* have in common.

Let W be a $m \times m$ term similarity matrix,

$$W = XX^T$$

In the binary weighting model, element $w_{ij} = t_i^T t_j$ represents the number of documents in which term *i* and term *j* co-occur (appear together).

2.2 Dimensionality Reduction (Rank Reduction) Using SVD

After the construction of the occurrence matrix, LSA finds a low-rank approximation to the term-document matrix. There could be various reasons for these approximations:

1. The original term-document matrix is presumed too large for the computing resources; in this case, the approximated low rank matrix is interpreted as an approximation (a "least and necessary evil").
2. The original term-document matrix is presumed noisy: for example, anecdotal instances of terms are to be eliminated. From this point of view, the approximated matrix is interpreted as a ***de-noisified matrix*** (a better matrix than the original).

3. The original term-document matrix is presumed overly sparse relative to the "true" term-document matrix. That is, the original matrix lists only the words actually in each document, whereas we might be interested in all words related to each document—generally a much larger set due to synonymy.

As the rank lowering is expected to merge the dimensions associated with terms that have similar meanings.

Now, from the theory of linear algebra, there exists a decomposition of X such that U and V are orthogonal matrices and Σ is a diagonal matrix. This is called a singular value decomposition (SVD):

$$X = \mathrm{U}\Sigma V^T$$

The matrix products giving us the term and document correlations from equation 2 and 3 then become:-

$$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = (U\Sigma V^T)(V\Sigma^T U^T) = U\Sigma\Sigma^T U^T$$

$$X^T X = (U\Sigma V^T)^T(U\Sigma V^T) = (V\Sigma^T U^T)(U\Sigma V^T) = V\Sigma^T\Sigma V^T$$

Since $\Sigma\Sigma^T$ and $\Sigma^T\Sigma$ are diagonal we see that U must contain the eigenvectors of XX^T, while V must be the eigenvectors of $X^T X$. Both products have the same non-zero eigenvalues, given by the non-zero entries of $\Sigma\Sigma^T$, or equally, by the non-zero entries of $\Sigma^T\Sigma$. Now the decomposition looks like this:

$$\underset{}{\overset{X}{\underset{(\mathbf{t}_i^T)\rightarrow}{}}}\overset{}{\underset{(\mathbf{d}_j)\downarrow}{}}\begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,n} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,j} & \cdots & x_{m,n} \end{bmatrix} = \overset{}{\underset{(\hat{\mathbf{t}}_i^T)\rightarrow}{}}\overset{U}{\begin{bmatrix} \begin{bmatrix} \\ \mathbf{u}_1 \\ \\ \end{bmatrix} \cdots \begin{bmatrix} \\ \mathbf{u}_l \\ \\ \end{bmatrix} \end{bmatrix}} \cdot \overset{\Sigma}{\begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_l \end{bmatrix}} \cdot \overset{V^T}{\underset{(\hat{\mathbf{d}}_j)\downarrow}{}}\begin{bmatrix} [& \mathbf{v}_1 &] \\ & \vdots & \\ [& \mathbf{v}_l &] \end{bmatrix}$$

Figure 8:- **Singular Vector Decomposition**

The values $\sigma_1 \ldots . \sigma_l$ are called the singular values, and $u_1 \ldots . . u_l$ and $v_1 \ldots . v_l$ are the left and right singular vectors. It turns out that when you select the k largest singular values, and their corresponding singular vectors from U and V, you get the rank k approximation to X with the smallest error. This approximation has a minimal error. But more importantly, we can now

treat the term and document vectors as a "semantic space". They are a lower-dimensional approximation of the higher-dimensional space.

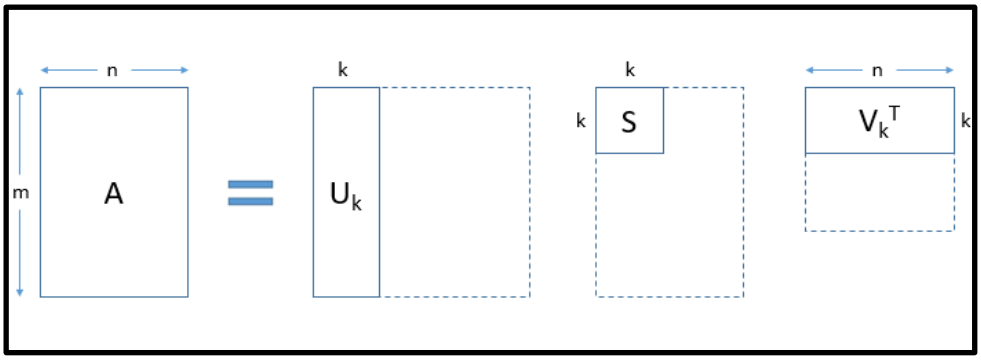

Figure 9:- **Truncated SVD**

The claim is that the similarity between documents can be computed more reliably in the low dimensional latent semantic space, since the redundancy and noise of the original term-document space is removed. Also, term associations into 'concepts' are captured since documents sharing frequently co-occurring terms have a similar representation in the latent space.

Applications of Topic Modeling:-

1. Finding similar documents across languages(cross language information retrieval).
2. For expanding feature space of machine learning and text mining.
3. Used for email filtering and spam detection.
4. Used by the education department in essay grading, text summarization etc.
5. Used in search engine optimization.
6. **LSIgraph.com** is the website which uses latent semantic analysis for the entered query keyword.

3. Report

3.1 Present Investigation

We scraped the data from 20 newsgroups and obtained news in form of text data. But the data was not clean. After cleaning that data, the text data was preprocessed in terms of removing all the stop words (‘a’, ‘an’, ‘the’, etc.), synonyms and those words which don't have any meaning. There were ***11,314*** rows in the dataset. The document term matrix was created where the rows are the terms and the columns represent the documents. The dimensions of the matrix are ***11314 x 73392*** . To show the relationship among document, the graph was plotted in 2 dimensions in order to visualize the dataset.

As we took the dataset from 20 newsgroups, the truncated SVD technique is applied on the data in order to reduce the dimensionality to 20 i.e. the top 20 singular values of the $\sum$ matrix was chosen. Because of this LSA technique all the noise from the data was minimized and each document was clustered as shown in UMAP. All the terms get inserted to their respective clusters, which was the purpose of the Topic Modelling.

3.2 Example of text data: Titles of Some Technical Memos

Let’s take an example of Titles of some Technical Memos

Titles	Text
C1	Human machine interface for ABC computer applications
C2	A survey of user opinion of computer system response time
C3	The EPS user interface management system
C4	System and human system engineering testing of EPS
C5	Relation of user perceived response time to error measurement
M1	The generation of random, binary, ordered trees
M2	The intersection graph of paths in trees
M3	Graph minors IV: Widths of trees and well-quasi-ordering
M4	Graph minors: A survey

Table 1:- Represents Technical Memos

3.2.1 Procedure to Perform LSA

Step1) Make a document term matrix of the given text data X

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

Table 2:- **Document-Term matrix**

A word by context matrix, X, formed from the titles of five articles about human-computer interaction and four graph theory. Cell entries are the number of times that a word(rows) appear in a title(columns) for that word.

If we represent the document and the terms in the 2D plane, then it will look like the following:-

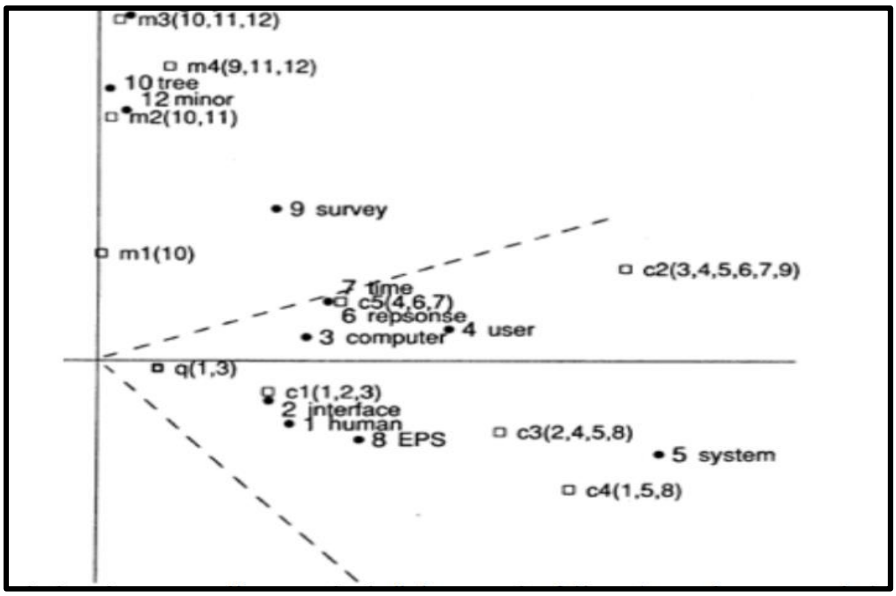

Figure 10:- **Representing document and term on 2D plane**

Step 2) Dimension Reduction of the document term matrix in such a way that the words occurring in some context now appear with greater or lesser estimated frequency, and some that did not appear originally do now appear, at least frequency. As there were **two types of contents** in this example, So, only two columns were selected for the truncated SVD. These two columns represent the number of topics. Applying SVD on matrix X,

$$X = \mathrm{U}\Sigma V^T$$

So, the U = 12 x 2 matrix, $\sum$ = 2 x 2 matrix and V^T = 2 x 9.

U	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
interface	0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
computer	0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
user	0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
system	0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
response	0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
time	0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
EPS	0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
survey	0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
trees	0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
graph	0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
minors	0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

Table 3:- **U matrix of SVD**

Σ	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	3.34								
interface		2.54							
computer			2.35						
user				1.64					
system					1.50				
response						1.31			
time							0.85		
EPS								0.56	
survey									0.36
trees									
graph									
minors									

***Table 4:- Σ* matrix of SVD**

V	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
interface	-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
computer	0.11	-0.50	0.21	0.57	-0.51	0.19	0.19	0.25	0.08
user	-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
system	0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
response	-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
time	0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
EPS	-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
survey	-0.06	0.24	0.02	-0.08	-0.26	-.062	0.02	0.52	-0.45
trees									
graph									
minors									

***Table 5:-* V matrix of SVD**

This is the final matrix obtained after doing the dimension reduction by considering that there are only two topics:-

$\hat{X}$	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.66	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Table 6:- Final Matrix Output

As shown in the matrix, each term has a value between [-1,1]. Greater the value in each cell, greater is its probability for that word to appear in that context.

Two-dimensional reconstruction of the original matrix shown in Table 1 based on shaded columns and rows from SVD as shown in Table 2. Comparing shaded and boxed rows and cells of Table. 1 and 4 illustrates how LSA induces similarity relations by changing estimated entries up or down to accommodate mutual constraints in the data.

3.2.2 Conclusion from This Example

1. Look at the two shaded cells for survey and trees in column M4. The word tree did not appear in this graph theory title. But because m4 did contain graph and minors, the zero entry for tree has been replaced with 0.66, which can be viewed as an estimate of how many times it would occur in each of an infinite sample of titles containing graph and minors.

2. By contrast, the value 1.00 for survey , which appeared once in M4, has been replaced by 0.42 reflecting the fact that it is unexpected in this context and should be counted as unimportant in characterizing the passage.

3. Very roughly, in constructing the reduced dimensional representation, SVD, with only values along ***two orthogonal dimensions*** to go on, has to estimate what words actually appear in each context by using only the information it has extracted.

4. This text segment is best described as having so much of abstract concept one and so much of abstract concept two, and this word has so much of concept one and so much of concept two, and combining those two pieces of information (by vector arithmetic), my best guess is that word X actually appeared 0.6 times in context Y.

4. Result and Conclusion

Topic modeling output is not entirely human readable. One way to understand what the program is telling you is through a visualization but be sure that you know how to understand what the visualization is telling you. Topic modeling tools are fallible, and if the algorithm isn't right, they can return some bizarre results.

In our investigation, the frequency of each newsgroup is shown in the following graph where the X-axis depicts the newsgroup and the Y-axis shows the count of topics.

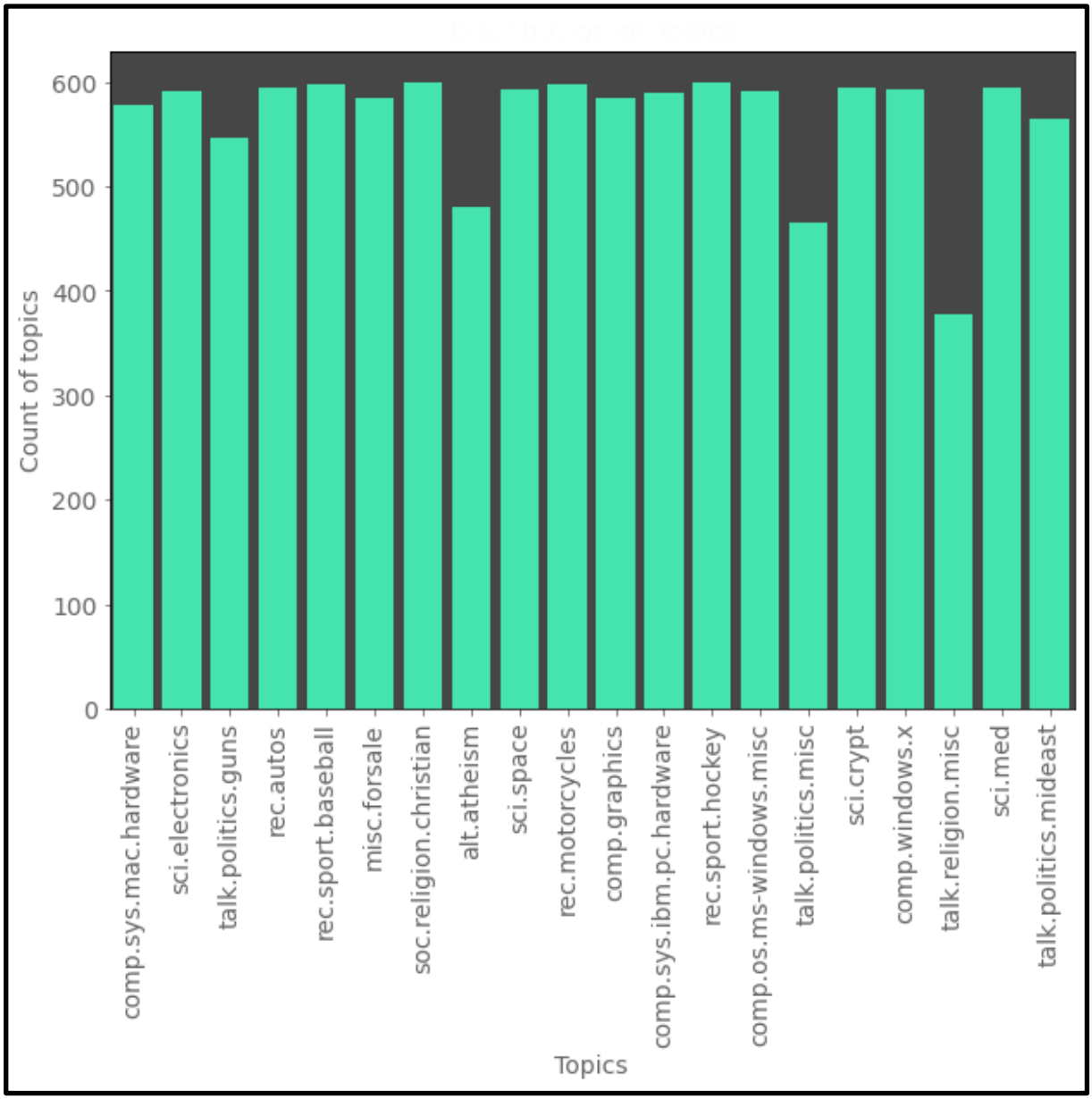

Figure 11:- Graph representing frequency of topic by each news group

The most common and weighted words before cleaning the dataset

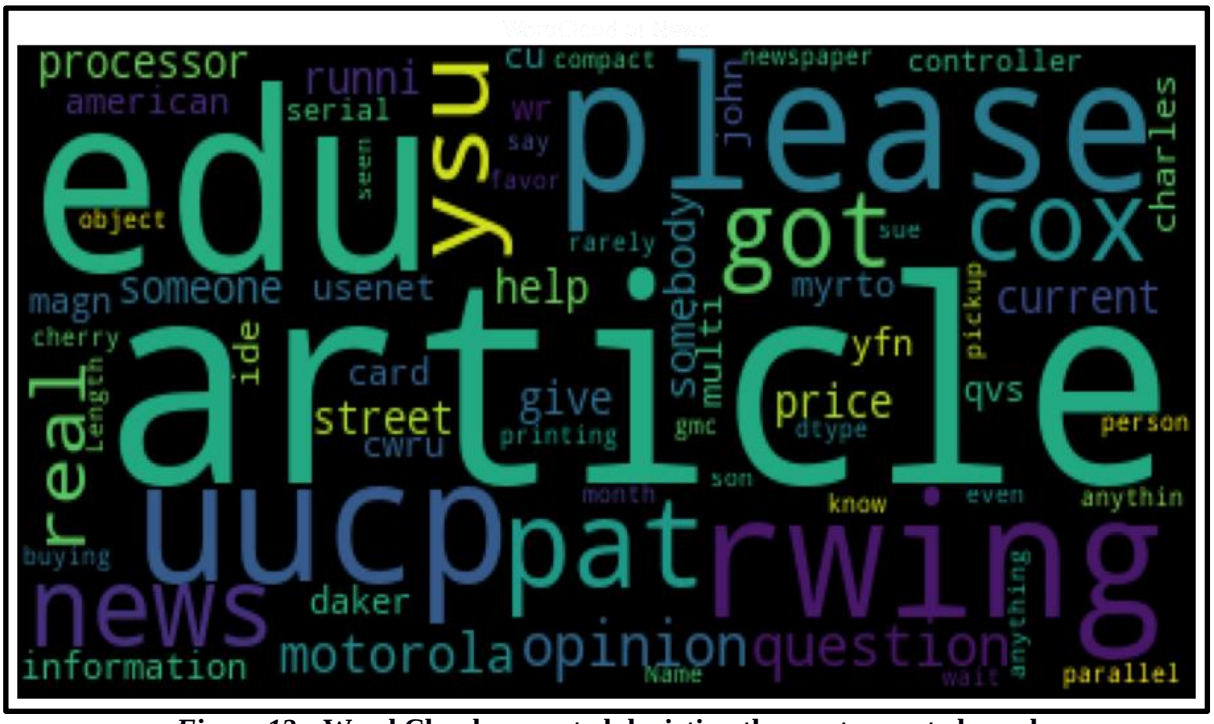

Figure 12:- **Word Cloud generated depicting the most repeated word**

When we try to visualize the news vector without any clustering algorithm in 2 dimensions, we get the following result:-

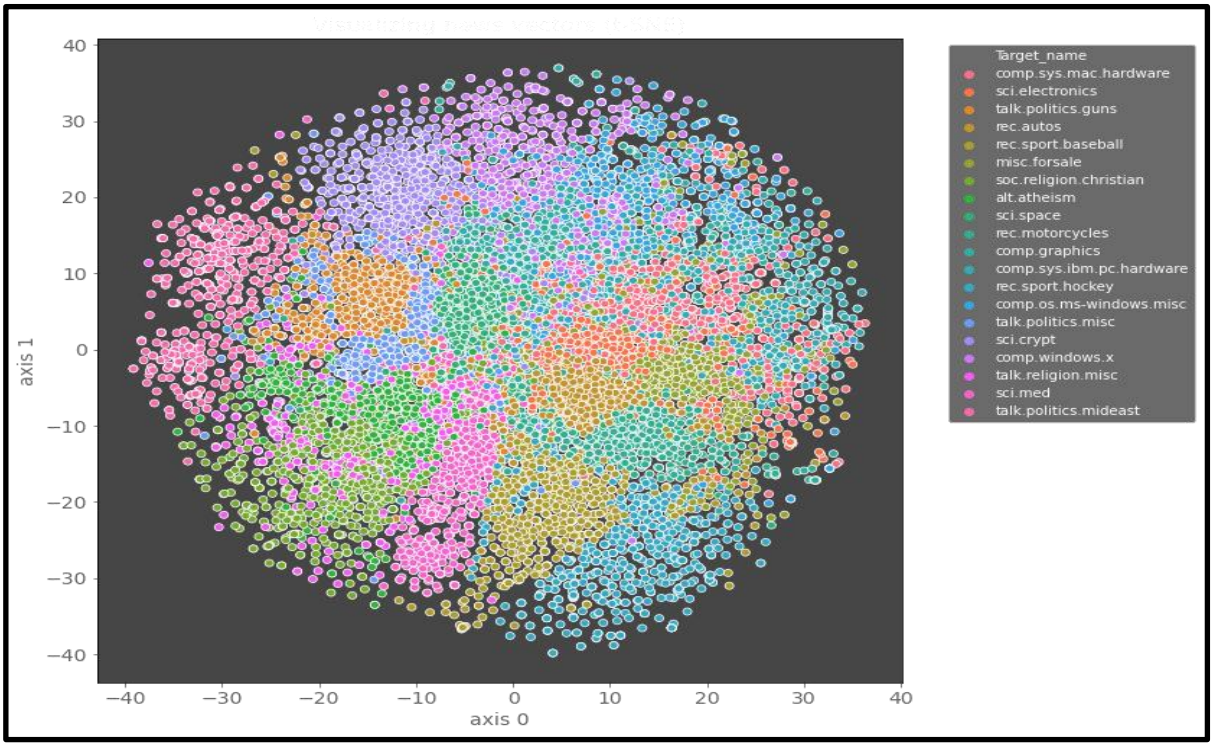

Figure 13:- **Vectorizing all the topic on Vector Space Model**

We clustered all the documents into 20 topics/concepts. Each topic contains the following term, which are co-related,

Topics	Words
Topic 1	edu would one writes com people article know like get
Topic 2	window drive card file thanks do driver disk program scsi
Topic 3	god window jesus christian bible file faith christ belief atheist
Topic 4	key chip encryption clipper government escrow system phone law algorithm
Topic 5	drive scsi ide controller disk hard floppy card bus god
Topic 6	game god key team player chip jesus hockey play season
Topic 7	edu window com writes article drive key scsi file apr
Topic 8	israel window israeli armenian drive arab gun jew people team
Topic 9	car window gun driver com card bike problem thing good
Topic 10	card driver video monitor israel color israeli bus bit vga
Topic 11	gun people firearm weapon com crime law card criminal scsi
Topic 12	space armenian nasa image turkish system gov launch orbit year
Topic 13	armenian com turkish armenia turkey turk thanks please genocide azeri
Topic 14	com israel israeli netcom irq sandvik port arab mouse modem
Topic 15	file car card driver com ftp god format image disk
Topic 16	scsi know anyone file thanks would bit bike driver bus
Topic 17	scsi space window driver nasa car card gov ide bus
Topic 18	scsi car mac mhz speed ide bus gun data device
Topic 19	system objective morality moral livesey file people keith value car
Topic 20	car key gun objective color bit morality drive value number

Table 7:- **Clustering all the terms which are related to same topic**

The UMAP of the text data after we apply LSA on the dataset. Here each dot represents a document and the color represents one of the 20 newsgroups.

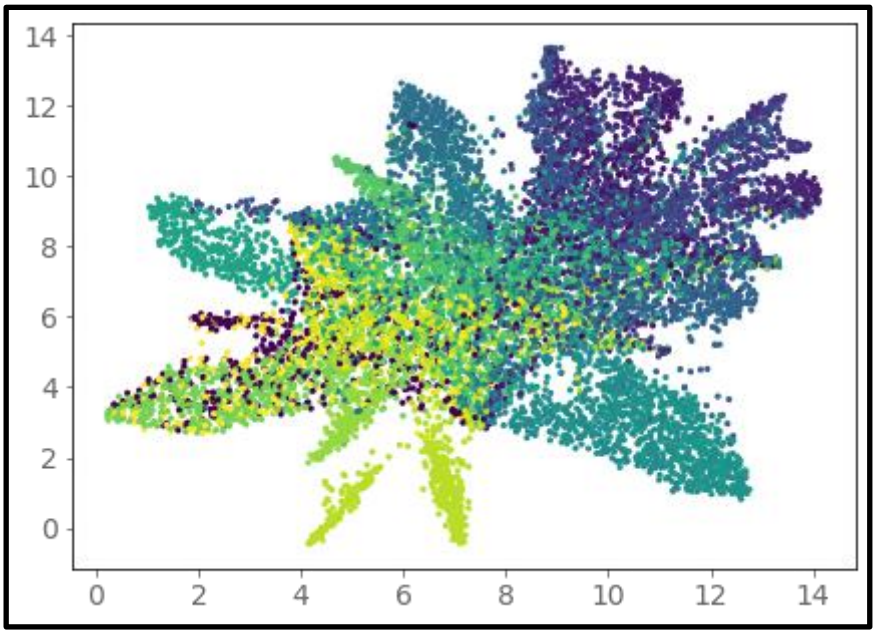

Figure 14:- **UMAP depicting the clustering of the documents**

5. Future Work

The following describes some future prospects for Topic Modeling- LSA:-

1. Experimenting on large dataset, because reducing the dimensions sometimes leads to loss of quality information.
2. The present work is carried out using tf-idf as the term weighting measure in the vector space model. Instead, experimental evaluation can be carried out to study the influence on the document structure by considering various supervised term weighting as done in Latent Dirichlet Allocation (LDA) which is more advance technique for Topic Modeling.
3. To apply Topic Modeling on Languages other than English, to get the latent (hidden) meaning of words in that language.

6. References

1. Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, Walter Kintsch, Handbook of Latent Semantic Analysis, 2013 edition,Chapter 2 ,by Hoboken : Taylor and Francis, 2013.
2. Landauer, Thomas; Foltz, Peter W.; Laham, Darrell (1998). "Introduction to Latent Semantic Analysis"
3. Berry, Michael; Dumais, Susan T.; O'Brien, Gavin W. (1995). "Using Linear Algebra for Intelligent Information Retrieval"
4. Deerwester, Scott; Dumais, Susan T.; Furnas, George W.; Landauer, Thomas K.; Harshman, Richard (1990). "Indexing by Latent Semantic Analysis" (PDF). *Journal of the American Society for Information Science*.
5. Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, Walter Kintsch, Handbook of Latent Semantic Analysis, 2013 edition,Chapter 2 ,by Hoboken : Taylor and Francis, 2013.
6. Anderson, J. R. (1990). The adaptive character of thought. Hillsdale, NJ: Lawrence Erlbaum Associates.
7. Anglin, J. M. (1970) The growth of word meaning. Cambridge, MA.: MIT Press.
8. Anglin, J. M., Alexander, T. M., & Johnson, C. J. (1996). Word learning and the growth of potentially knowable vocabulary. Submitted for publication.
9. Berry, M. W. (1992). Large scale singular value computations. International Journal of Supercomputer Applications, 6, 13-49.

10. Berry, M. W., Dumais, S. T. and O'Brien, G.W. (1995) Using linear algebra for intelligent information retrieval. SIAM: Review, 37, 573-595.
11. Britton, B. K. & Sorrells, R. C. (1998/this issue). Thinking about knowledge learned from instruction and experience: Two tests of a connectionist model. Discourse Processes, 25, 131-177.
12. Burgess, C., Livesay, K. & Lund, K. (1998/this issue). Explorations in context space: Words, sentences, discourse. Discourse Processes, 25, 211-257.
13. Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society For Information Science, 41, 391-407.
14. Dumais, S. T. (1994). Latent semantic indexing (LSI) and TREC-2. In D. Harman (Ed.), The Second Text Retrieval Conference (TREC2) (National Institute of Standards and Technology Special Publication 500-215, pp. 105-116).
15. Dumais, S. T. & Nielsen, J. (1992). Automating the assignment of submitted manuscripts to reviewers. In N. Belkin, P. Ingwesen, & A. M. Pejtersen (Eds.) Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, Association for Computing Machinery.
16. Harman, D. (1986). An experimental study of the factors important in document ranking. In Association for Computing Machinery Conference on Research and Development in Information Retrieval. Association for Computing Machinery.
17. Kintsch, W. (1988) The role of knowledge in discourse comprehension construction-integration model. Psychological Review, 163-182.
18. Laham, D. (1997a). Automated holistic scoring of the quality of content in directed
19. student essays using Latent Semantic Analysis. Unpublished master's thesis, University of Colorado, Boulder.