**Abstract:**

On the internet there are numerous documents available for any single search, some of those contain the information that we want, some of the contain a mixture of multiple topics and some might contain very less amount of information that we want, finding it manually is a very time consuming job, in order to find out which topic that particular document is related to and in what percentages we introduce three topic modelling techniques, LDA, LSA, PLSA and we determine which one has better performance.

**Keywords:** Topic modelling, LDA, LSA, PLSA.

**1.Introduction**:

In many domains, unstructured data, like PDF files, web pages, and online articles, is increasing rapidly. Analysing these unstructured data and drawing some meaningful conclusions from these data,  is becoming a challenging task. To solve this challenging task, a powerful technique called topic modelling is employed. It is an unsupervised ML technique, that can uncover underlying themes, structures, and patterns from the unstructured documents without any prior knowledge. There are three widely used methods for topic modelling: PLSA, LSA and LDA. All these three methods, LSA, LDA and PLSA use statistical algorithms to identify patterns and relationships within the text, enabling the extraction of significant topics from unstructured data.

Probabilistic latent semantic analysis (PLSA) makes use of probabilistic methods to take care of the issue. Discovering the probabilistic model with hidden topics which can create the data that we notice in the term-document matrix is the main aim of PLSA.

Latent Semantic Analysis makes use of an algebraic technique called singular value decomposition for identifying latent semantic patterns in a term-document matrix. We convert the raw text into a document-term matrix using two methods, the first one uses a bag of words and the second one uses tf-idf. LSA identifies topics by recognizing the latent dimensions that contribute maximum to the variance in the term-document matrix. By projecting the documents onto these dimensions, LSA assigns topic weights to each document and word weights to each topic.

On the other hand, LDA is a generative probabilistic model which thinks that a document is represented as a combination of topics and each topic is identified by a distribution of words. Here, we assume that a document is a combination of different topics, every word within the document can be assigned to one or more of these topics. By using topic-word and document-topic distributions, LDA assigns the most probable topics for the documents and the most probable words to the topics.

In our research, we focus on modelling unstructured documents using both LDA and LSA and comparing them. First, we will begin with preprocessing the unstructured text data, making sure that it is in a suitable format for further processing. Next, we will implement LDA using Gibbs sampling and LSA using SVD. Our objective is to make our methods as accurate as possible by fine-tuning the models and optimising the parameters to achieve meaningful and reliable topic extraction results.

In this paper, Section 2 contains the related work, and Section 3 discusses the proposed methodology, workflow, and preprocessing steps involved in our work, Section 4 discusses the technology used like different methods, software, and hardware used, in our work; Section 5 discusses the results and analysis of our work, and Section 6 concludes and the whole work with the future scope of our research.

**2. Related Works:**

Research is a very important domain for the advancement of mankind as well as for stakeholders and researchers to invest their energy and money in specific areas that show development. There are numerous works on topic modelling and information retrieval using latent Dirichlet Allocation, LSA, and PLSA in multiple areas using different methods. Rahul Kumar Gupta et al. [[1]](https://www.mendeley.com/catalogue/4bf60dec-c215-32d0-acd5-977a707cbc23/) proposed work on the prediction of research trends in 2022. They have experimented using LDA with BoW and TF-IDF scores to catch new directions and paradigms. In this research work, web crawling is used to collect data from articles and pre-process the data by removing unimportant information. Then they used LDA topic modelling to extract latent topics from the corpus. They created a baseline model using Bag of Words and then improved upon it by incorporating TF-IDF scores. The model showed a 41% increase in accuracy compared to the baseline model, but there were some limitations, such as the need for human involvement in topic labelling and the inability to directly model the correlation between topic occurrences. The authors plan to include the evolution of a system that can automatically analyse trends in future work. Despite the limitations, the proposed model performed well in terms of coherence score without involving deep learning or graph networks.

India is one of the world’s biggest newspaper markets, with approximately 100,000 newspaper channels and 1,300,000,000 readers, so it is very important to find the important topics that belong to the same type of news articles that are in two separate languages and classify them for this using LDA. Anukriti Srivastav et al. [[2]](https://www.mendeley.com/catalogue/11e74628-8e45-361c-abc9-5f9a7b6a1784/) proposed a work for the context-based topic identification of English and Hindi news articles in 2021. This research aims to extract core topics from two different news articles that are written in Hindi and English, and then measure the similarity score between them. The approach involves scraping trending news articles from Google News feeds for both languages, translating the Hindi articles to English using Google Translator, preprocessing the data by tokenizing, removing stopwords, and creating bigrams. The research uses the LDA to find best topics and calculate coherence scores for each topic. The dataset used is a news corpus collected from Google News. The coherence score decides the optimal number of topics for each language, which is 42 for English and 40 for Hindi. The research shows that LDA with cosine similarity performs better when compared to Doc2Vec in measuring the similarity between news articles.

In the medical sector, information increases rapidly, so it should be stored using an appropriate structure so that we can retrieve it easier from the corpus in which the information related to the medical sector is stored. For this reason, M. Selvi et al. [[3]](https://www.mendeley.com/catalogue/7e9d7402-3e9e-3bdd-8ef7-25e73174ed6a/) proposed work for the classification of the medical dataset in 2019. This paper proposes an intelligent medical data mining system that uses machine learning classification algorithms to effectively store and retrieve medical data with relevancy and is tested on breast and liver datasets collected from the University of California, Irvine repository. It uses Naive Bayes, Support Vector Machine (SVM), Differential Topic Modelling (DTM) with LDA and SVM, and a newly proposed method called Naive Bayes with NBTC. The system proposed here includes a user-friendly interface for displaying the mined results with graphical output. Here, LDA-based topic modelling is used for feature extraction and to classify the topics in the dataset. The NBTC classifier provides better classification accuracy compared to existing algorithms. Word cloud analysis is used for effective storage and visualisation of the dataset. Overall, the proposed system carries out better decision-making when compared to existing systems.

Many industries, like social media, platforms for printing news, and online searches, are some of the crucial platforms that are facing overhead of textual data. Here, the input data is not in a structured format and contains vast amounts of important information that needs to be supervised and handled precisely. To decrease such a large amount of data, Hritvik Gupta et al. [4[]](https://www.mendeley.com/catalogue/6b7fcf4a-14ba-3fc0-a94e-8fd3bf209249/) proposed work for text summarization and sentence-based topic modelling in 2021. This research paper introduces a method for generating an extractive text summary using LSA topic modelling with truncated singular value decomposition (SVD). The suggested LSA model extracts and illustrates the relevant meaning of words for computing the similarity between sentences. It uses tf-idf to analyse text documents and realise topics by doing matrix decomposition on a document-term matrix with the help of SVD. Kaggle’s news dataset is used for both text summarization and evaluation. They experimented over multiple sets of documents and are divided into seven stages, which include using BERT for encoding sentences and TF-IDF keyword extraction to extract keywords. The cosine similarity is calculated between the sentence and topic by using positional embeddings that are generated by BERT. The algorithm sorts and selects the top five sentences with the highest cosine similarity to form the summary. Preprocessing of text is done by dividing the document into different sentences and then normalising, removing punctuation, stop words, and stemming.

Spoken document summarization plays an important role in the future network era, where digital content in multimedia, including speech information, will become key for retrieval and browsing. For this summarization, Sheng-Yi Kong et al. [[5]](https://www.mendeley.com/catalogue/49562658-2aea-3010-b622-2ec091f3b42c/) proposed work for summarising spoken documents in 2006. This research paper proposes new methods based on PLSA modelling of terms and documents for identifying the most important sentences of the topics that are conveyed by the spoken documents. This paper employed two PLSA-based statistical measures, based on topic significance and word entropy, and compared them with the well-known significance score. Broadcast news stories in Mandarin Chinese were used for the experiment. The experiment's results showed that, for 10% and 30% summarization ratios, the term entropy is better compared to commonly used significance scores. Although often performing worse than the other two measures, the proposed topic significance achieved a comparatively stable performance for the 50% summarization ratio. This paper also discusses the challenges of automatic summarization for spoken documents, such as error recognition, speech problems, and a lack of correct sentences.

In the present network era, email has emerged as one of the most significant media for communications. So for qualitatively investigating the interaction and influence among email users, Dong Zhang et al. [[6]](https://www.mendeley.com/catalogue/7d59b21b-6c31-3101-8237-4847e0b3a26d/) proposed work for modelling interactions from email communication in 2006. This paper proposes a framework for investigating email interactions and influence among users using a dynamic Bayesian network (DBN) and probabilistic latent semantic analysis (PLSA) on the Enron email dataset. In these experiments, the emails that were received by one of the 150 users are used. The proposed framework uses the semantic content of emails to model user interactions and employs a clustering algorithm to discover the community structure of an organisation. The influence model in the DBN models interacting Markov chains at the individual user and interaction levels. The proposed framework has several parts, including email parsing, text preprocessing, PLSA language modelling, and influence modelling. Limitations of this work include a lack of comparison with other methods and a lack of comprehensive evaluation. However, the proposed model can be easily applied to a multimedia email corpus, and future work could investigate higher-order Markov models to handle some emails that invalidate the assumption.

**3.Proposed Methodology**:

The following methodology is used for generating top the n words of topics in a corpus of documents: The idea is to find which method gives the better result. For this, we first implemented LDA for our preprocessed data using Gibbs Sampling. As a second step, we implemented LSA for the same data. In this method, we find the document-term matrix using tf-idf scores, and for this matrix, we apply SVD. In the last step, we implemented PLSA using the EM algorithm. The complete flow diagram of a proposed model is shown in Fig. 1.

**3.1. Data collection**

For implementing our three methods, we collected four different text documents: the first document is taken on rivers; the second document is taken on the city of Mumbai; the third document is taken on global warming and the fourth document is based on  Mumbai: a city of dreams. For these documents, we are applying LDA, LSA, and PLSA.

**3.2. Pre-processing**

In this preprocessing stage, we used several NLP techniques to remove irrelevant information from the unstructured data before training and further analysis. The following steps are followed for preprocessing the user selected documents

**Lowercasing**: The collected data is transformed into a uniform case by converting all text to lowercase. This helps in standardising the text and avoids duplication of words based on case variations.

**Punctuation Removal:** As some word embedding models do not support punctuation marks, so  they are removed. Removing punctuation ensures that text remains compatible with the chosen models.

**Tokenization:** Tokenizing involves dividing the text into smaller units which are called tokens. This process helps in breaking down the text into meaningful components such as words or subwords and facilitates further analysis.

**Stop Word Removal:** involves removal of the words that do not considerably add to the overall meaning of the context. After the removing tokens, stop-words are removed.

Because of stop word removal we mainly have two benefits that is

1.Reduced data size

2.Narrowing down the feature space improves performance

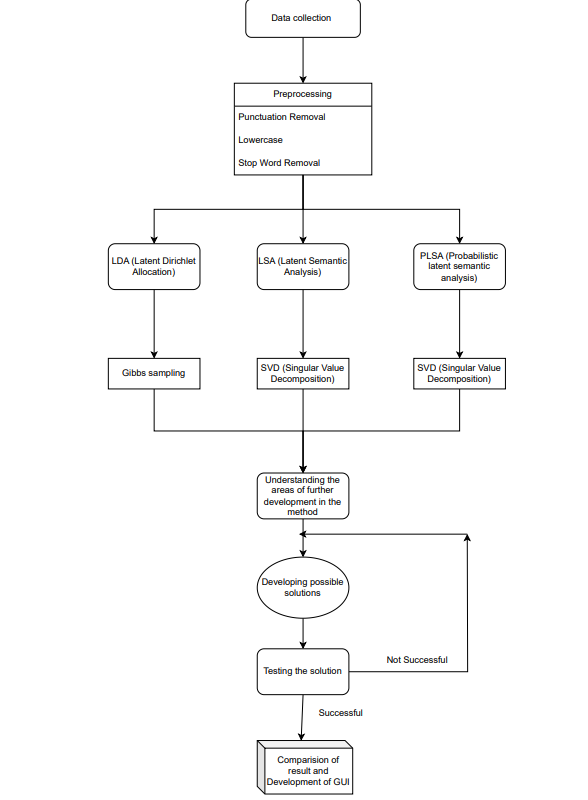


                                                                   Fig. 1. WorkFlow

**4. Technology**

**4.1. LDA**

A ML technique called topic modelling is used to find words and uncover patterns within the document. Topic modelling aims to uncover the main topics within the documents and generate a collection of related words. The first model that we employed for topic modelling is Latent Dirichlet Allocation.

LDA is an unsupervised ML technique that is used for probabilistic clustering. Here, the assumption is that in a corpus, each document is composed of different topics, and every topic is a combination of multiple words. LDA’s primary objective is to uncover the hidden topic structure in a collection of documents. By calculating the frequency with which certain words appear together in a document, LDA identifies which words are more closely related to each other. The input data is divided into two parts by this model.

1. The percentage of each topic in the document
2. Word occurrence probability for each topic

The LDA model is effective because it captures the competing sparsity between the topic distribution of documents and the word distribution of topics.

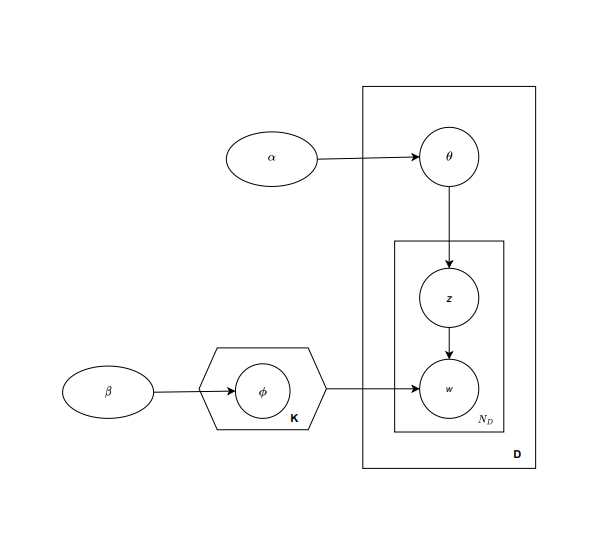


                                                                  Fig.2. Blueprint of LDA model

In Fig.2. K represents quantity topics, ND represents quantity words in the document, D represents the quantity of documents to analyse, w represents words, , are dirichlet distributions, W,Z are multinomials.

The probability of a document is calculated as follows:

P(W,Z,,;,)=k=1MP(k;)l=1NP(l;)t=1UP(Zk,t|k)P(Wk,t|Zk,t)                  (1)

The two important parameters in LDA include and parameters.

parameter: It is a Dirichlet prior parameter that indicates document-topic density; if is large the documents contain multiple topics and lead to a greater focused topic distribution per document.

While the parameter indicates topic-word density, if is high, the topics are assumed to be made up of most of the words, which results in a more specific word distribution per topic.

There are two types of dirichlet distributions, the first connects documents with the corresponding topics, and the second associates topics with the corresponding words. After the preprocessing part is done, we calculate the document-term matrix, where the rows of the matrix are documents and the columns contain words from the corpus.

**4.1.1 Gibbs sampling**

After creating a document-term matrix, the next step is topic modelling using LDA. It is required to define the K topics prior to executing the method. Our main aim is to choose K, which allows for identifying understandable topics. Here we should take care that the generated topics should not be too high due to the unclear and complicated analysis of the results. In our case, we are taking from the set 3,4,5. "Gibbs sampling is a simulation tool for obtaining samples from a non normalised joint density function." [7]. Gibbs sampling is started randomly at a point; we will be mentioning the number of iterations we want. In our case, we checked our result with 100 iterations, as well as with 1000 iterations but as the number of iterations increased, the computation time also increased as well as the accuracy of the result also got increased.

**4.2. LSA**

LSA is also a topic modelling technique; the aim of Latent Semantic Analysis is to represent every document and term in a high-dimensional space, and then we decrease the dimensions of the space and extract the underlying latent semantics using SVD. In LSA, text document collection is represented as a term-document matrix, in which rows are the document and columns are the words having some score. The main goal of LSA is to capture the underlying semantic relationships between terms and documents,even in the presence of sparsity. The main goal is to consider a term-document matrix and decompose it into two different matrices.

1. The first one is the document-topic matrix.
2. The second matrix is the topic-term matrix.

After the preprocessing step, we create a term-document matrix of dimension m x n, where columns represent a word having some score to calculate these scores. There are several methods. One of those methods is to fill each column entry with a raw count of the number of times a particular word (suppose jth) appears in a particular document (ith). But in practice, this method does not work as they do not focus on the significance of each word in the document so for this, we use another method called TF-IDF.

**4.2.1. TF-IDF**

The tf-idf approach is used to weigh terms because it assigns a value to a term based on its importance in a single document, scaled by its importance across all the documents.

In TF-IDF, term frequency (TF) specifies how many times a specific word appears in the document among the number of times all words appear in the same document.

While IDF specifies how common a word is among all the documents, it is calculated as follows:

tfidf(i, d, D)=tf(t,d)idf(t,D)                                                                                 (2)

It seems that if there exists a term t' that is present in all documents, then 0 is given as the idf score, so we are eliminating t' from the tf-idf corpus, which results in purifying the words that directly impact the results. After generating a document-term matrix, we realised that our matrix is very noisy, sparse, and redundant in multiple dimensions. But to find hidden topics that capture the connections between words and documents, we need to do dimension reduction on our term-document matrix.

**4.2.2.** **Singular Value Decomposition (SVD)**

For performing the dimension reduction on the generated document-term matrix, we used the SVD technique. The reason for using SVD is to find the most valuable information. SVD factors any matrix into a combination of three different matrices. Out of the three decomposed matrices, the first is V, which is a column matrix; U is a row matrix; and both U and V are orthogonal. The third matrix is which is a singular matrix and contains singular values as its diagonal elements. By selecting the t largest singular values, where t is the hyperparameter, the truncated SVD reduces the dimension .

S=UVT                                                                                                                              (5)

In the matrix U, rows determine document vectors, while rows in matrix V are term vectors. The rows in the document-topic matrix represent a vector corresponding to the document, and the vector representation for the terms can be found in the term-topic matrix. SVD returns a vector representation for every document and term in our data as an output. Using these vectors, we find similar words and documents.

**4.3. Probabilistic Latent Semantic Analysis (PLSA)**

In PLSA, instead of using SVD like in LSA, we use probabilistic methods to solve the problem, and the basic assumptions are the same as in LSA and LDA, which are discussed in previous sections.

In the latent variable model for PLSA, we first sample the documents, then, based on the documents, we sample topics, and then, based on topics, we sample words. That means here documents and words are conditionally independent given a hidden topic. In this framework, we focus on finding the relationship that exists between observed and hidden variables. PLSA adds a probabilistic spin to the basic assumption in topic modelling.

1. The probability of  a topic present in the document is p(z|d).
2. The probability of  a Words taken from the topic is p(w|z).

Using these two probabilities, we find that the joint probabilities, p(w|z) and p(z|d) are modelled as multinomial distributions and are trained using  the EM algorithm.

**4.3.1. EM (Expectation-Maximization)algorithm**

We have two steps in the EM algorithm. The first step is the E-step, where using observed available data, we guess missing data values. In the M-step, we use the complete data that is in the E-step and update the parameters. We repeat these steps until we reach the solution. The fundamental goal of the EM algorithm is to update the parameter values in the M-step with the help of observed data from our dataset and eliminate the missing data for the latent variables.

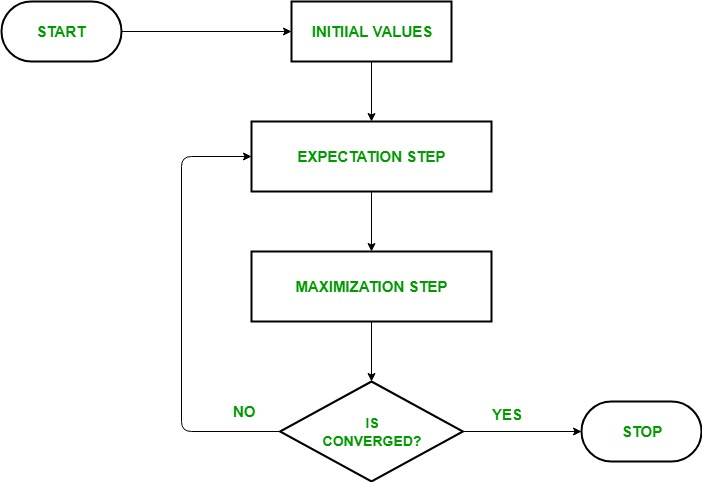


                                                                Fig.3. Flow of EM algorithm [8]

**4.4. Software and Hardware required**

The most important aspect of this project is the software, which was developed using Python (version 3.10). We have developed the codes for the processes of LDA and LSA and enabled the user to perform all of the topic modelling operations by providing them with the functions of the methods discussed. Our primary goal was to make the software as fast and responsive as possible, for which we have made use of certain libraries, which will be discussed later.

We have chosen the Python programming language because it can be easily used to write the complex logic of LDA and LSA without much hassle, and there is also a lot of support for different operations in the libraries present in Python.

We have also made use of the tkinter library in Python in order to develop a GUI-based interface so as to facilitate topic modelling for non-technical people and make it accessible to a larger audience. We have made a simple .exe file that can be shared with different users and installed to perform topic modelling. The GUI is very interactive and does not have any learning curve, and the users can easily perform the actions without needing any manuals. There is an option to select the folders containing the text files on which the topic modelling can be performed, and then we have given the users the option to select from LDA and LSA techniques. They can also choose the number of topics into which they want to classify the given files. The users can select a large number of topics, and the GUI works fine for that.

Now let us understand the hardware requirements for our software. As our software does not require much CPU power, any CPU above the Intel Pentium will work fine. The amount of RAM needed is dependent on the size of the input corpus of the users, and at least 4GB of RAM is recommended. For using the software in the form of a .exe file the user must have a Windows operating system where the software can simply be accessed by double clicking the icon. The user can simply make the required selections and go about selecting the method that they want to use in order to perform topic modelling on their input data by clicking on the execute button.

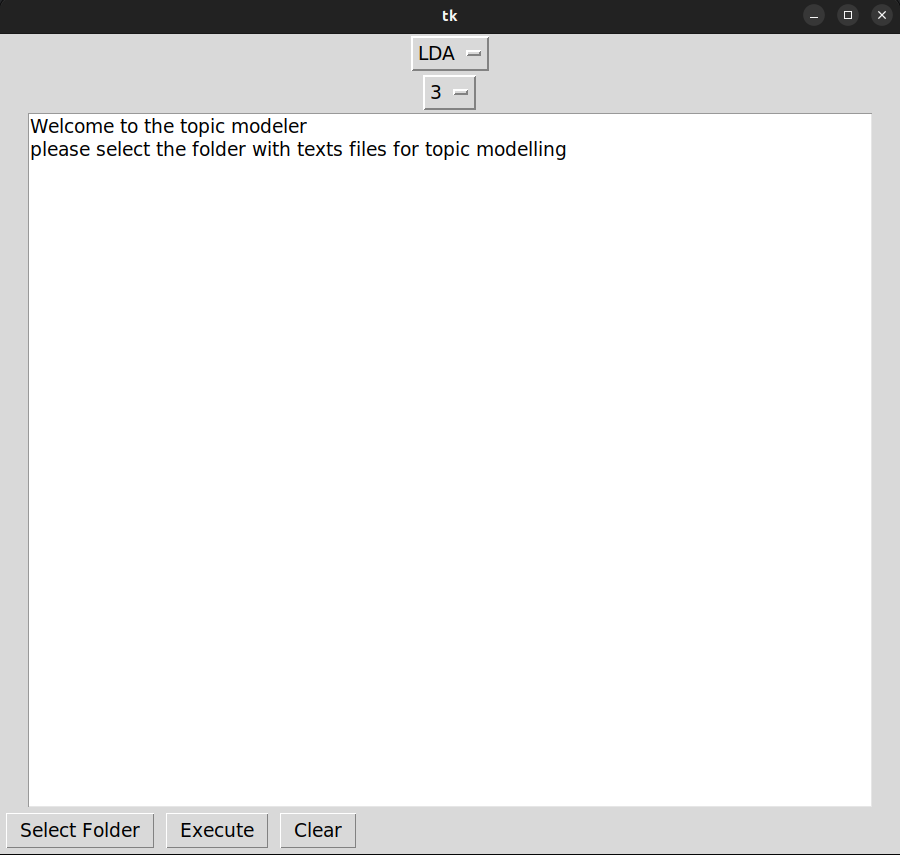


Fig.4. GUI of the Topic Modelling Software

**5. Results & Analysis**:

We have written a Python code for LDA and tested it on various essays on various different topics, the results we received was satisfactory if given enough number of iterations to the algorithm for Gibbs sampling, a few similar topics were also used for testing the effectiveness of the algorithm, if it classifies them into the same topic or not and the results were successful.

The topics chosen were:

Essay 1: Rivers

Essay 2: City of Mumbai

Essay 3: Global Warming

Essay 4: Mumbai: a city of dreams

The results that we got are at par with the actual labels of the topics which are evident in the percentages for each topic in each document shown in the figure below.

The results of our topic modelling techniques are included in form of a screenshot:

When we analysed the results we came to know about the performance and the speed of these algorithms, on making a comparison we can say that the LDA algorithm takes the maximum amount of time but yields very good results from which we can infer the topics very easily, the reason for this is the Gibbs sampling part which takes a lot of time in order to allocate the values using dirichlet distribution.

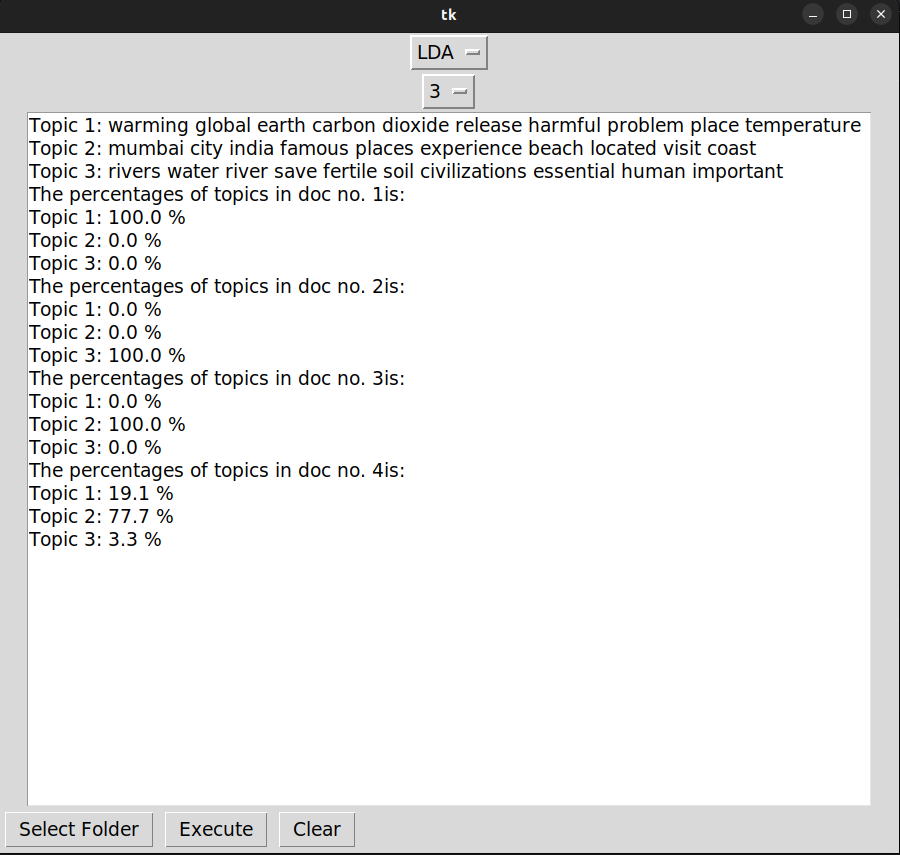


Fig.5. Results for LDA

On the other hand we have the LSA which takes very less amount of time to execute but the results that we get from it are not as interpretable as in the case of LDA or PLSA, that is due to the process of SVD which has to select the maximum variance components out of the various dimensional corpus,

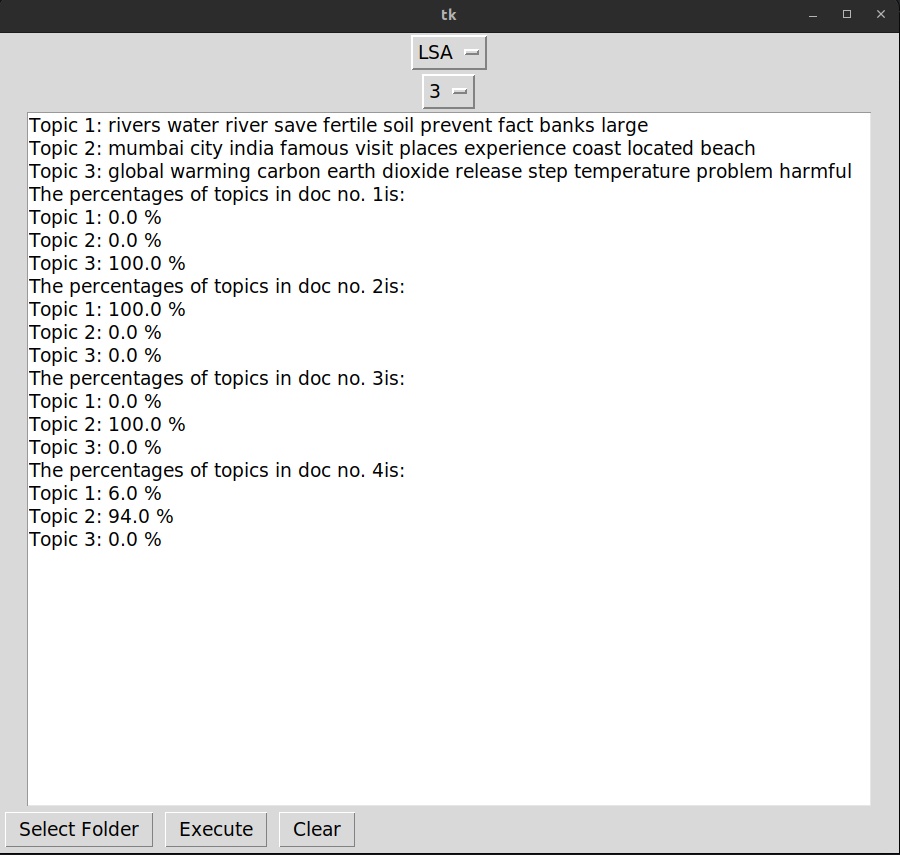


                                                            Fig.6. Results for LSA

Now if we consider PLSA we observe that the amount of time is less in comparison with LDA but the performance is almost at par this is because in PLSA there is no requirement of setting of the values on the basis of dirichlet distribution which helps to make it much faster, also the use of expectation maximisation algorithm makes it faster than LDA.

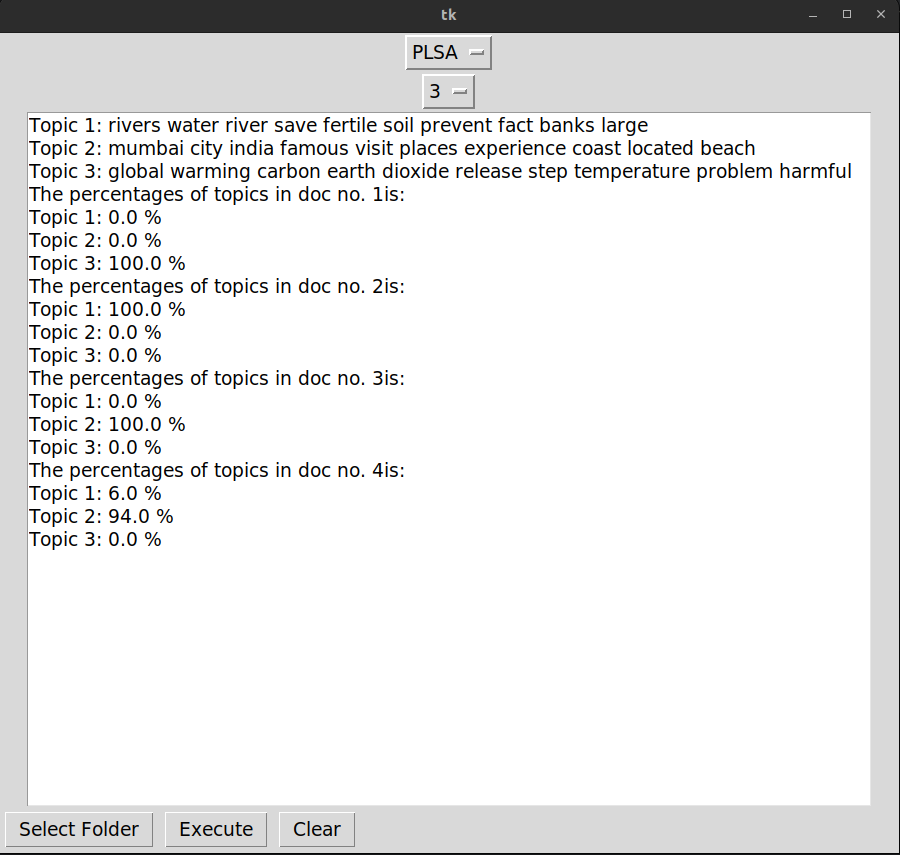


                                                    Fig.7. Results of PLSA

We have included all of these algorithms in the GUI tool that has been created by us which allows the user to select and apply any algorithm which meets their requirements.

**6. Conclusion and Future Prospects:**

In conclusion after conducting all of this research and experiments we observed that LDA and PLSA are the superior techniques out of the three, although a bit time consuming, they still provide very accurate results and their topics can be interpreted easily, LSA on the other hand is inaccurate but provides the results very fast and could be useful in cases where time is an important factor to be considered.

Now about the future prospects of this project, there has not been much work in the research field of the topic of PLSA so we could make the PLSA algorithm even better and the GUI tool which we have developed can include more features in the upcoming versions, also they can be integrated through the browser so that we can perform topic modelling on large online documents and can get a clear idea of what is useful for us and what is not so useful.