A Project Report on

# Unstructured Text Analysis

Submitted in partial fulfillment of the requirements for the degree of Computer Engineering in Bachelor of Engineering –VIII


**Submitted by:**

Aalap Jethwa: 140173107014

Ankit Singh: 130170107108




Department of Computer Engineering
Vishwakarma Government Engineering College,
Chandkheda, Ahmedabad


**2016-2017**

# DECLARATION

This is to certify that

i) The project comprises my original work towards the degree of bachelor of Engineering in Computer Engineering at Vishwakarma Government Engineering College, Chandkheda, under the Gujarat Technological University, Ahmedabad and has not been submitted elsewhere for a degree.

ii) Due acknowledgement has been made in the text to all other material used.

Aalap Jethwa
Ankit Singh

# C E R T I F I C A T E

This is to certify that the project having title "Unstructured Text Analysis", towards the fulfillment of the requirements for the degree of Bachelor of Engineering in Information Technology of Vishwakarma Government Engineering College, Chandkheda, under the Gujarat Technological University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

**Under the Guidance of**

**Internal Guide**

Signature:

Name: Prof. Uttam G. Chauhan

Designation: Assitant professor of Computer Eng.

Organization: VGEC, Chandkheda

**Head of Department**

Prof. M.T.Savaliya,

Associate Professor

**Computer Engineering Department,**

**Vishwakarma government engineering college, Chandkheda, Ahmedabad.**

# ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us during this period. This project would not have been materialized without the help from many quarters. We sincerely thank to all the persons who ever played a vital role in the successful completion of Analysis of our project **"Unstructured Text Analysis"**.

We express our deep sense of gratitude to Head of Department (Computer Department) **prof. M.T Savaliya**, Our Internal Guide **Prof. Uttam sir**, who has given us the opportunity to work with them. Their guidance, suggestions and expertise have been a source of inspiration and were very helpful to me during my tenure.

We are grateful to "Vishwakarma Government Engineering College, Chandkheda," for giving us some official important information and guidance for making this project. We are also thankful to the Whole staff of Computer Department under whose guidance we completed our Analysis task for project, who devoted their precious time. In spite of, there in busy schedules they always came forward to guide us in our work whenever needed.

Finally, our deepest acknowledgement to our family and friends for their companionship, love and support.

Aalap Jethwa
Ankit Singh

# INDEX

# 1.Introduction

## 1.1 General Overview:

The opportunity cost of any business to ignore unstructured data is paramount in today's fierce competitive world. According to an IDC survey, unstructured data takes a lion's share in digital space and approximately occupies 80% by volume compared to only 20 for structured data. While the unstructured data is available in abundance, the number of software products and solutions that can accurately analyze the text, present insights in an understandable manner along with the ability to integrate such insights readily into other extant models that use numerical only data are rare. A lot of the challenges in this space arise from the fact that natural language provides the flexibility to convey exactly the same meaning in umpteen different ways, or worse, exactly the same statement in a different context may convey completely different meaning. Machine learning algorithms designed to analyze numerical data exactly know the structure of numbers and they are pre-programmed to process the data with precision. In case of natural language, it gets very problematic. Dialects, jargon, misspellings, short forms, acronyms, colloquialism, grammatical complexities, mixing one or more languages in the same text are just some of the fundamental problems unstructured data poses. It makes extremely difficult to precisely analyze unstructured data in the same way we process structured data.

## 1.2 Objective:

The objective is to develop information retrieval system which analyze large corpus of unstructured data in some useful patterns. The system can be applied to some domain of data to derive/infer useful data. Moreover, this system can be further extended for various regional languages to support information retrieval for a greater mass of people.

## 1.3 Technology Used:

To implement our project, the first and the foremost thing needed to bolster our project was a good LDA Topic Modeling set consisting of all relevant terms in the world to create our topic model. Initially we are testing on a small dataset but to increase our spectrum once we achieve reasonable result in this small dataset we are going to adopt large corpus. The programming language we are going to use to implement the various aspects of the project is ‘Python’.

# 2. Algorithm

## 1. Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes the following generative process for each document **w** in a corpus *D*:

1. Choose *N* Poisson(x).

2. Choose $\theta$ Dir($\alpha$).

3. For each of the *N* words $w_n$:

   (a) Choose a topic $z_n$ Multinomial ($\theta$).

   (b) Choose a word $w_n$ from $p(w_n|\ z_n,\beta)$ a multinomial probability conditioned on the topic $z_n$.

Several simplifying assumptions are made in this basic model, some of which we remove in subsequent sections. First, the dimensionality *k* of the Dirichlet distribution (and thus the dimensionality of the topic variable *z*) is assumed known and fixed. Second, the word probabilities are parameterized by a $k \times V$ matrix $\beta$where $\beta_{ij} = p(w^j = 1|\ z^i = 1)$, which for now we treat as a fixed quantity that is to be estimated. Finally, the Poisson assumption is not critical to anything that follows and more realistic document length distributions can be used as needed. Furthermore, note that *N* is independent of all the other data generating variables (q and **z**). It is thus an ancillary variable and we will generally ignore its randomness in the subsequent development. A *k*-dimensional Dirichlet random variable q can take values in the $(k-1)$-simplex (a *k*-vector q lies in the $(k-1)$-simplex if $\theta_i \geq 0$, $\sum_{i=1}^{k} \theta_i = 1$), and has the following probability density on this simplex:

$$p(\theta|\alpha) = \frac{\tau(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \tau(\alpha_i)} \theta_1^{\alpha^{i-1}} \ldots \theta_k^{\alpha^{k-1}}, \qquad (1)$$

where the parameter $\alpha$ is a *k*-vector with components $\alpha_i \geq 0$, and where $\tau(x)$ is the Gamma function.

The Dirichlet is a convenient distribution on the simplex—it is in the exponential family, has finite dimensional sufficient statistics, and is conjugate to the multinomial distribution. These properties will facilitate the development of inference and parameter estimation algorithms for LDA.

Given the parameters a and b, the joint distribution of a topic mixture q, a set of *N* topics **z**, and a set of *N* words **w** is given by:

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta)\, p(w_n|z_n, \beta) \ , \qquad (2)$$

**Figure 1:**

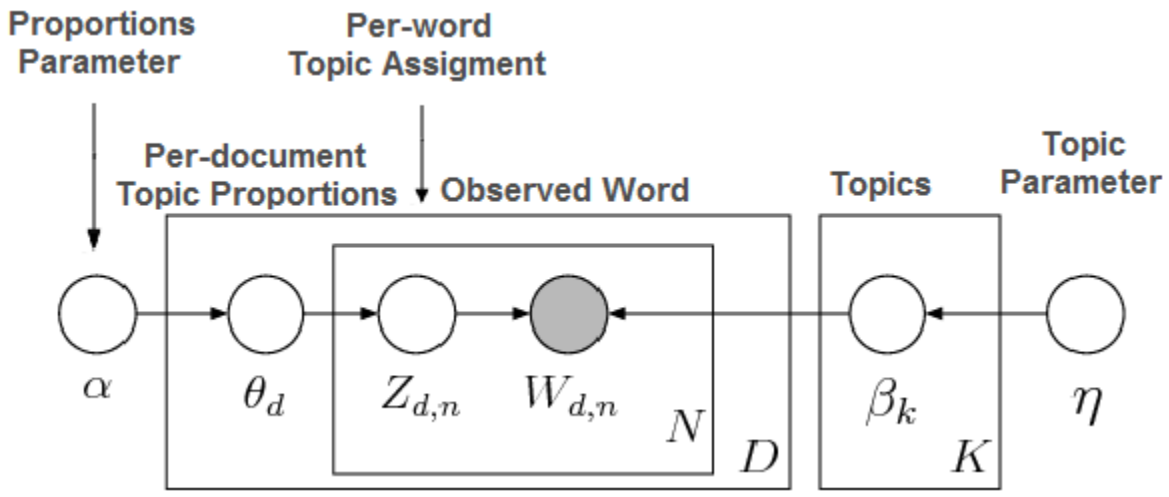



Figure 1: Graphical model representation of LDA. The boxes are "plates" representing replicates.

The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document.

where $p(z_n|\theta)$ is simply $\theta$ for the unique *i* such that $z_n^i = 1$ Integrating over q and summing

over z, we obtain the marginal distribution of a document:

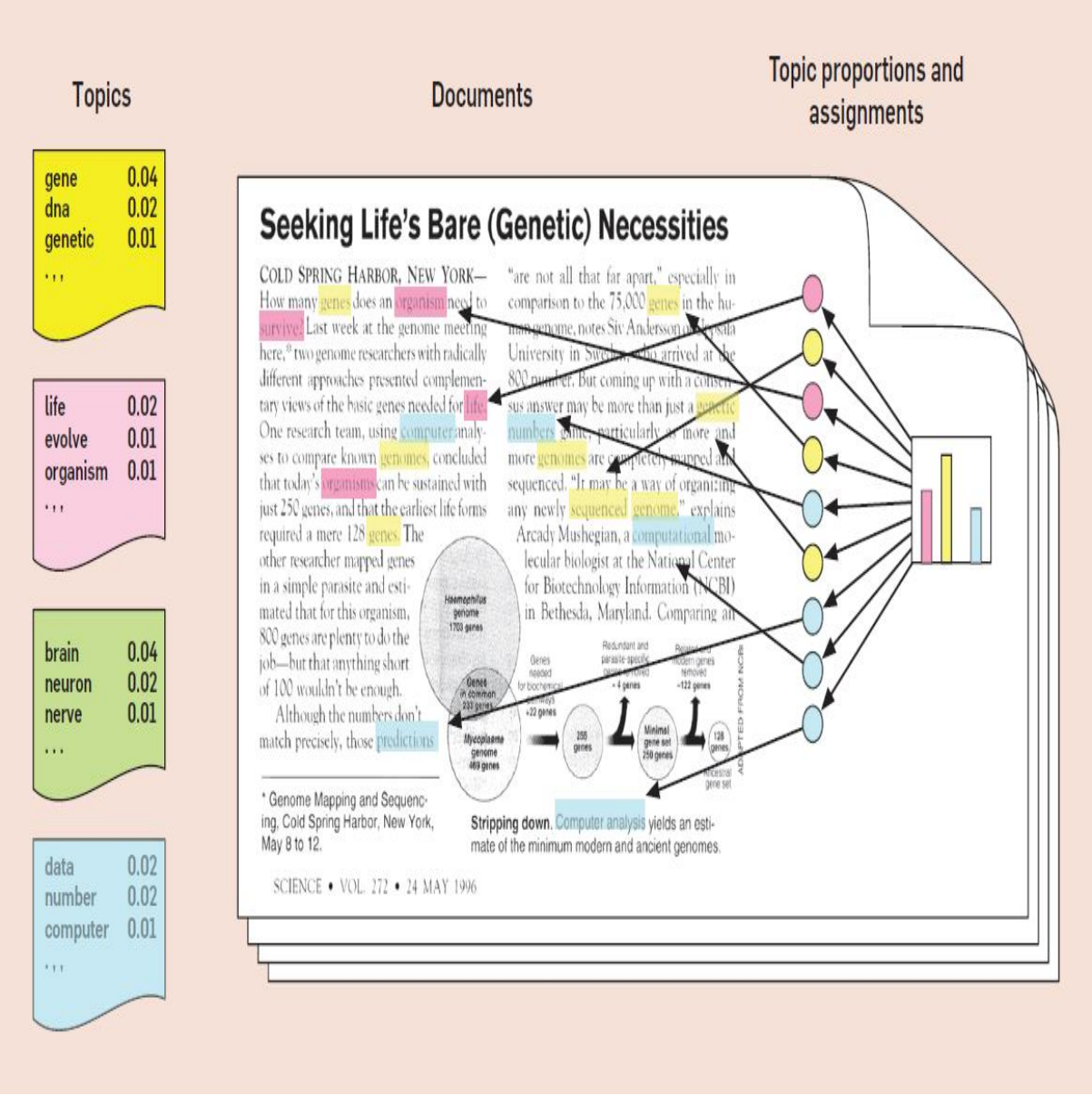
$$p(w|\alpha,\beta) = \int p(\alpha|\theta)\left(\prod_{n=1}^{N}\sum_{z_n} p(z_n|\theta)p(w_n|\, z_n,\beta)\right)d\theta \qquad (3)$$

Finally, taking the product of the marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D|\alpha,\beta) = \prod_{d=1}^{M}\int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}} p(z_{dn}|\theta_d)p(w_{dn}|\, z_{dn},\beta)\right)d\theta_d, \qquad (4)$$

The LDA model is represented as a probabilistic graphical model in Figure 1. As the figure makes clear, there are three levels to the LDA representation. The parameters a and b are corpus level parameters, assumed to be sampled once in the process of generating a corpus. The variables $\theta_{\mathrm{d}}$ are document-level variables, sampled once per document. Finally, the variables $z_{dn}$ and $w_{dn}$ are word-level variables and are sampled once for each word in each document. It is important to distinguish LDA from a simple Dirichlet-multinomial clustering model. A classical clustering model would involve a two-level model in which a Dirichlet is sampled once for a corpus, a multinomial clustering variable is selected once for each document in the corpus, and a set of words are selected for the document conditional on the cluster variable. As with many clustering models, such a model restricts a document to being associated with a single topic. LDA, on the other hand, involves three levels, and notably the topic node is sampled *repeatedly* within the document. Under this model, documents can be associated with multiple topics. Structures similar to that shown in Figure 1 are often studied in Bayesian statistical modeling, where they are referred to as hierarchical models or more precisely as conditionally independent hierarchical models. Such models are also often referred to as parametric empirical Bayes models, a term that refers not only to a particular model structure, but also to the methods used for estimating parameters in the model (Morris, 1983). Indeed, we adopt the empirical Bayes approach to estimating parameters such as a and b in simple implementations of LDA, but we also consider fuller Bayesian approaches as well.

**Figure 2:**


Topics
Documents
Topic proportions and assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK—
How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

## 2. Tf-Idf:

Tf-idf stands for *term frequency-inverse document frequency*, and the tf-idf weight is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the tf-idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query.

Tf-idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.

Typically, the tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), aka. the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

**TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (the total number of terms in the document) as a way of normalization.

**TF (t) = (Number of times term t appears in a document) / (Total number of terms in the document).**

**IDF: Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following:

**IDF (t) = log_e (Total number of documents / Number of documents with term t in it)**.

## 3. Gibbs Sampling:

As Bayesian models of cognitive phenomena become more sophisticated, the need for inference methods become more urgent. In a nutshell, the goal of Bayesian inference is to maintain a full posterior probability distribution over a set of random variables. However, maintaining and using this distribution often involves computing integrals which, for most non-trivial models, is intractable. Sampling algorithms based on Monte Carlo Markov Chain (MCMC) techniques are one possible way to go about inference in such models. The underlying logic of MCMC sampling is that we can estimate any desired expectation by ergodic averages. That is, we can compute any statistic of a posterior distribution as long as we have N simulated samples from that distribution:

$$\mathrm{E}[f(s)]p \approx \frac{1}{N}\sum_{i=1}^{N} f(s^{(i)}), \quad (6)$$

where P is the posterior distribution of interest, f(s) is the desired expectation, and f(s(i)) is the $i^{th}$ simulated sample from P.

- Algorithm of Gibbs sampling is as follow:

**Algorithm 1** Gibbs sampler

Initialize $x^{(0)} \sim q(x)$
**for** iteration $i = 1, 2, \ldots$ **do**
$x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$\vdots$
$x_D^{(i)} \sim p(X_D = x_D | X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \ldots, X_D = x_{D-1}^{(i)})$
**end for**

# 3. Literature Review:

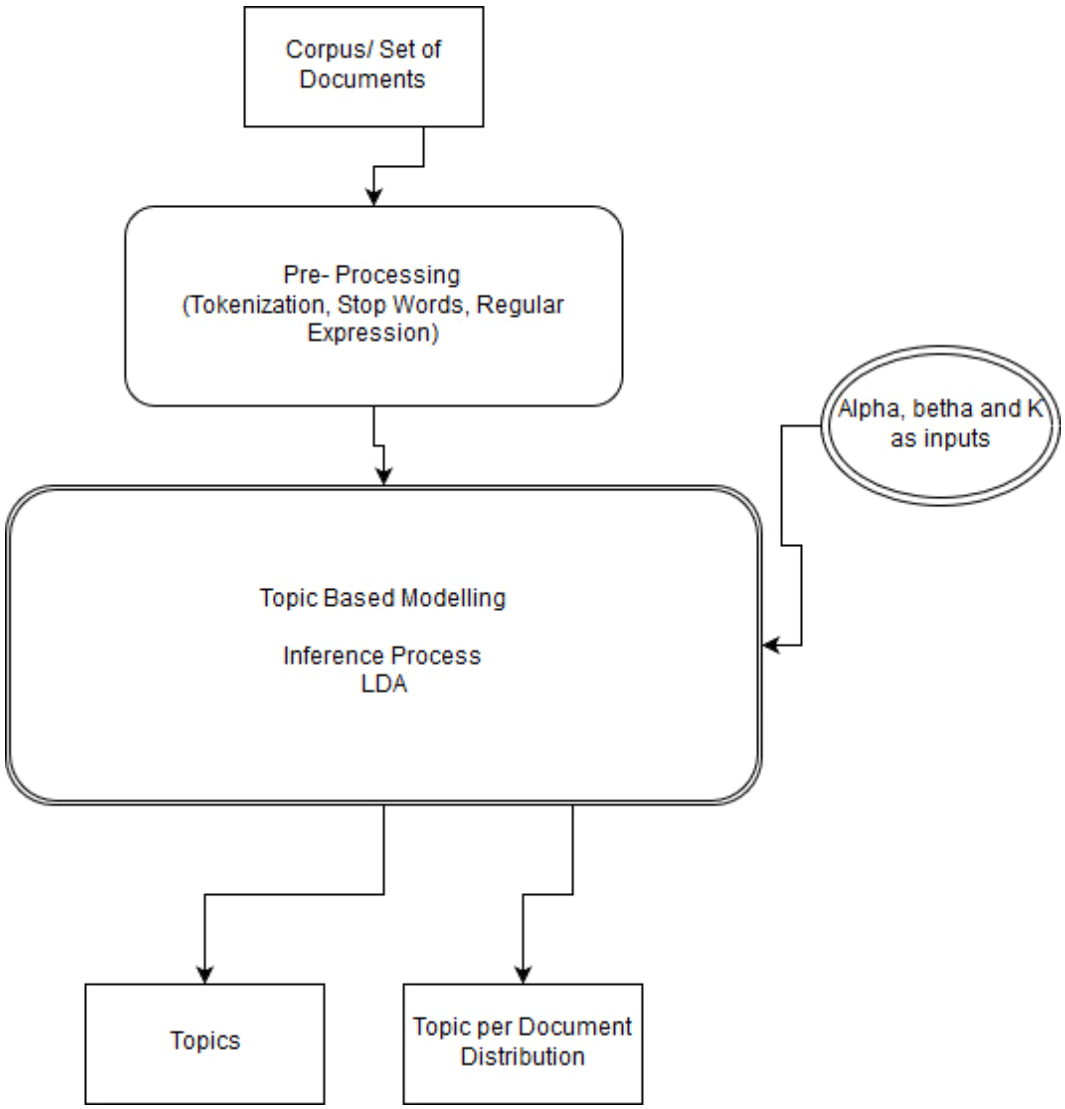
We referred many documents to divide the large corpus into relevant topics as well as to retrieve information from unstructured data. They are as follow:

- We used LDA to represent document as random mixtures over latent topics, referred from Latent Dirichlet Allocation By David M. Blei, Andrew Y. Ng and Michael I. Jordan. University of California, Berkeley.
- Gibbs Sampling to obtain a sequence of observations which are approximated from a specified multivariate probability distribution. Gibbs Sampling by Ilker Yildirim Department of Brain and Cognitive Sciences University of Rochester.
- To study Variational Inference we studied research papers of
    - Collapsed Variational Dirichlet Process Mixture Models by Max Welling of Dept. of Computer Science, UC Irvine.
    - Variational Dirichlet Process by David Blei and Michael I Jordan of Columbia University.
- To implement all algorithms, we preferred python and libraries like sckit-learn for machine learning and pandas for data mining. From web site - https://docs.python.org .

# 4. Approach

## 4.1 Initial Approach

Figure 3: Our initial approach to this project




Our system proposes a data driven approach toward the application. Using data from the Text files to store and preprocess it.

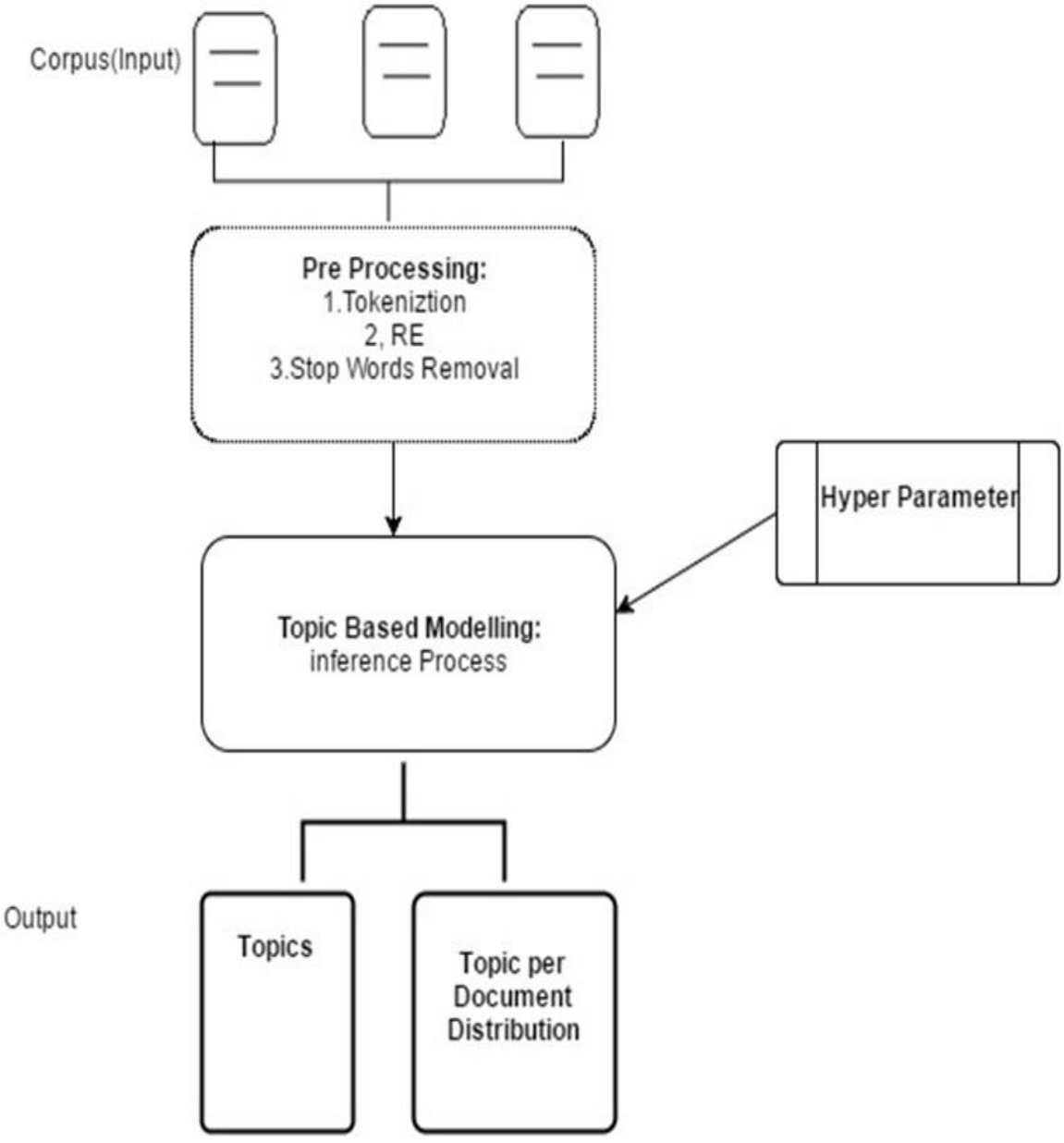
Here preprocess means to extract tags from the data to create a metadata associated with each document. This metadata will consist of the most potential tokens of the words which we compared and ranked against all the other tokens in the same genre to identify potential tokens unique to the topic and cluster to the genre and subgenres.

These tokens can be extracted using the TF-IDF approach. TF-IDF is the acronym of Term Frequency - Inverse Document Frequency, a standard approach used to identify keywords in a document. These keywords act as an abstract topic description of the movie and its plot. The abstract description can then be used to identify movies if the keywords are present in the natural language query.

The preprocessing of natural language query will be done using stemming and lemmatization. Stemming provides removal of stop words such as “that”,” it”, “a”, “an” etc. for quick identification of the important terms from the query. Lemmatization reduces the remaining words to its root so that a word like “describes” changes to “describe”. This allows us to reduce the overall unique keywords encountered. This can be performed on a bag-of-words model of query using the NLTK library of Python. Another approach to query preprocessing will be Named Entity Recognition. In Named Entity Recognition, the query is not considered as separate keywords but instead is taken as a term consisting of one or two words to represent an entity. This entity is given a class label which can be used to search the database of plots to find the appropriate result.

# 4.2 Architecture

Figure 4: Architecture of final approach

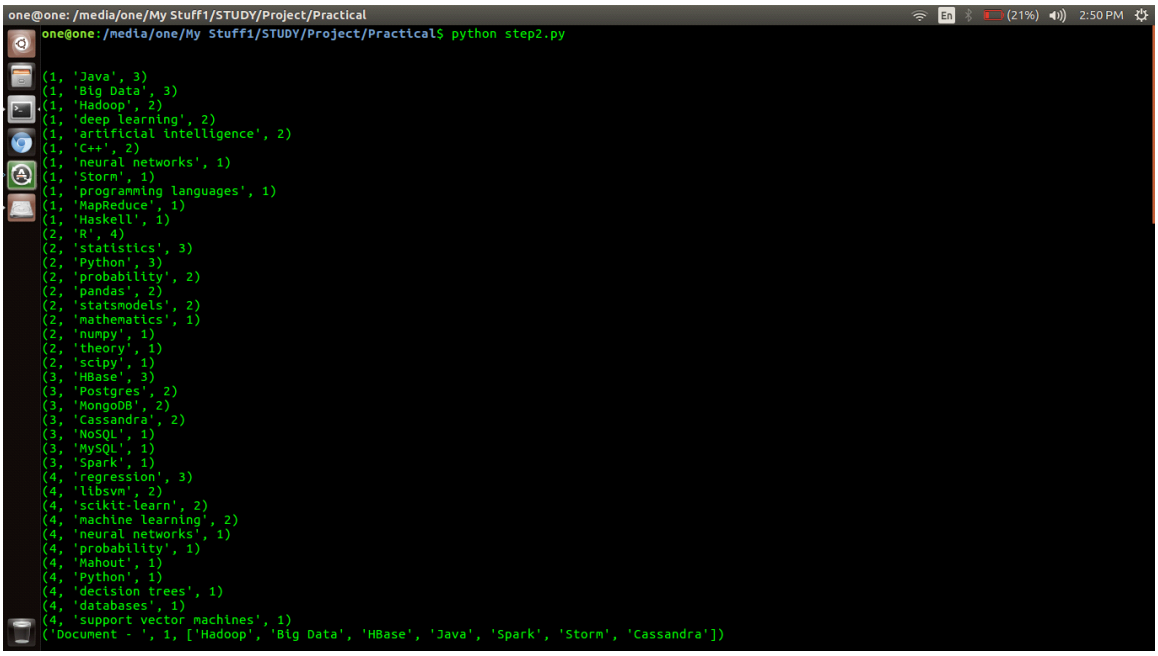


We built the system as mentioned above, using:

- Python: Python provide a development platform to quickly realize the algorithms with necessary speed and portability.
- Sckit-learn: The all in one Machine Learning package of Python provides very efficient implementations of all the required algorithms starting from Naive-Bayes till Tf-idf.
- NLTK: Natural Language Toolkit is the Python library implementing all the necessary text modelling and analysis features like stemming, lemmatization, chunking, named entity recognition
- Pandas: The Pandas Data Frame allows us to work with tables in Python with fast modification and inbuilt data analysis.

# 4.3 Results

Figure 5: Result after performing initial stage of algorithm



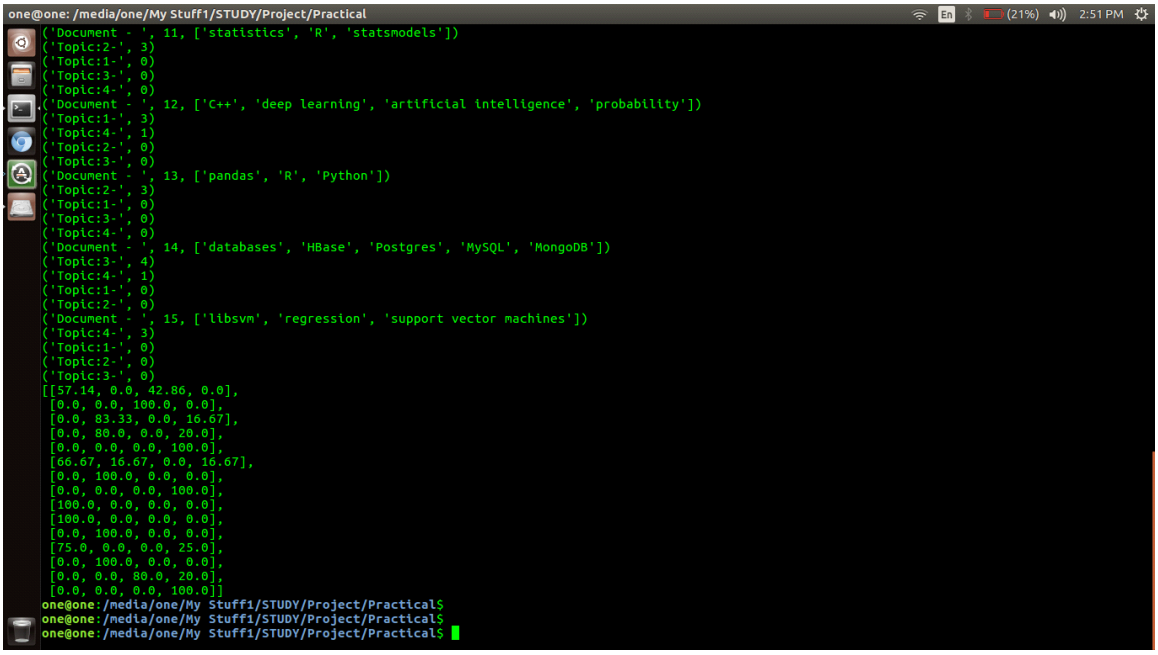



Figure 6: Various Topics created after the input was taken as text files

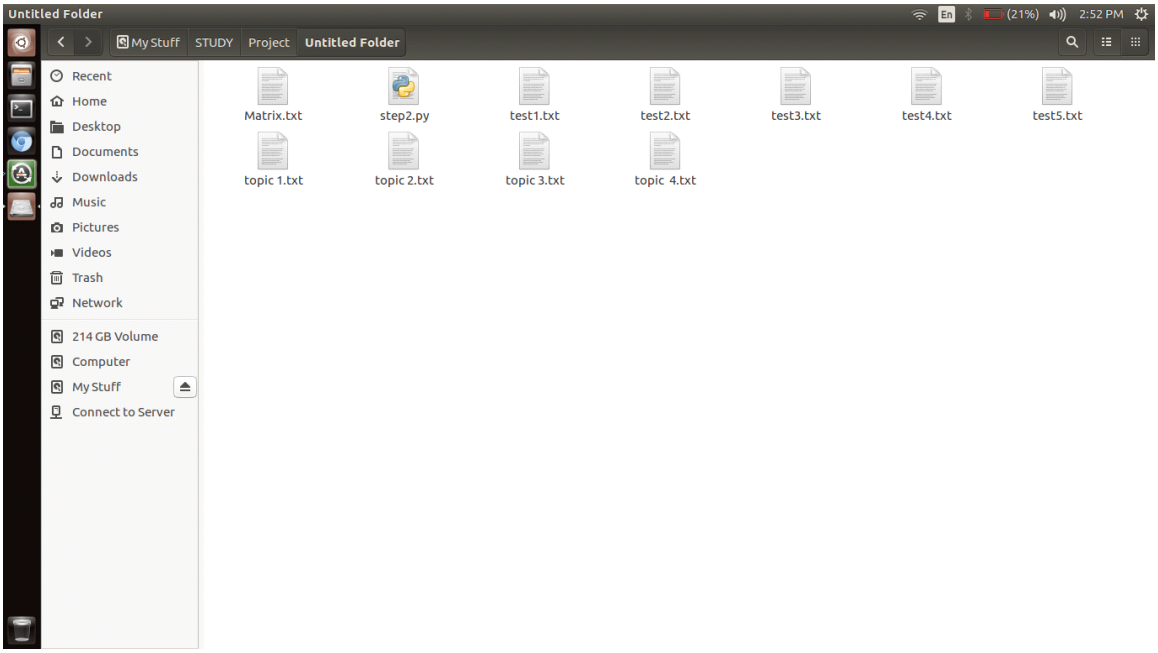
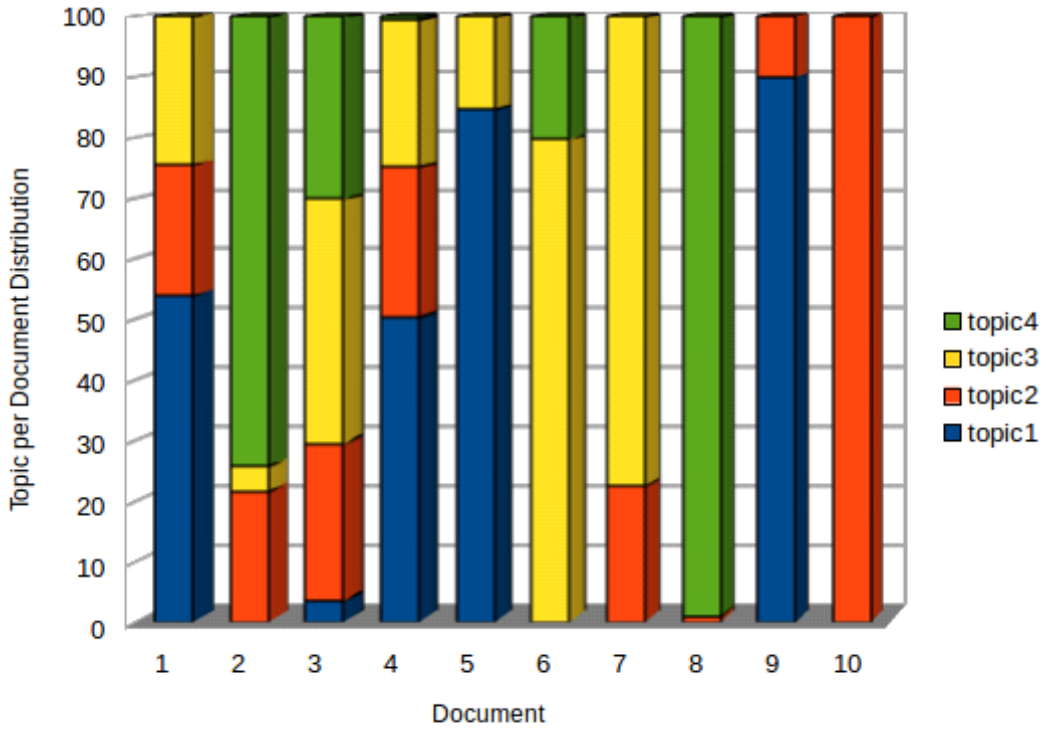



Figure 7: Graph showing the distribution of Topic per Document Matrix



# 5. Future Work

We want to train a parser to detect bigram and trigram words in the documents to get better embedding. This project does not limit us to one application. If the patterns and methods applied to all types of texts, we can use it to create a generic implementation to use this system across various regional languages and also extend our platform to a large mass of users. Another area is of Part of Speech (POS) which would be much helpful in novel or autobiography type documents.

# 6. Conclusion

With our final model using LDA, we have been able to represent the topic modeling in many contexts and capture the distributional semantics on various text files to enable efficient searching and retrieval of related documents in acceptable time. We have showed how word embedding are language and dictionary neutral. We hope to use this approach to make a scalable model and Topic modeling based on search query and get enough data to extrapolate unexplored topics of data clusters.

The need for natural language processing is more than ever. Combined with query processing and information retrieval, understanding a query is more useful than ever as it allows to go ahead the basic interaction mechanism of touch and type. This project will allow us to work very close to the domain of NLP and ML and it will provide us the platform to implement said algorithms to solve a problem in the real world.

With the refinement and the tweaking a few parameters here and there we are expecting to even get more good result. With the adoption of Topic based modeling and ultimately shifting to distributed architecture is a good omen for our system that we can conclude by the amount of time required to process the result on a single machine. Finally, to summary we can conclude that with some minor changes and a distributed approach we are bounded to achieve more accurate results.