**TOPIC MODELLING USING MACHINE LEARNING AND PYSPARK**

Minor project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

# Computer Science and Engineering

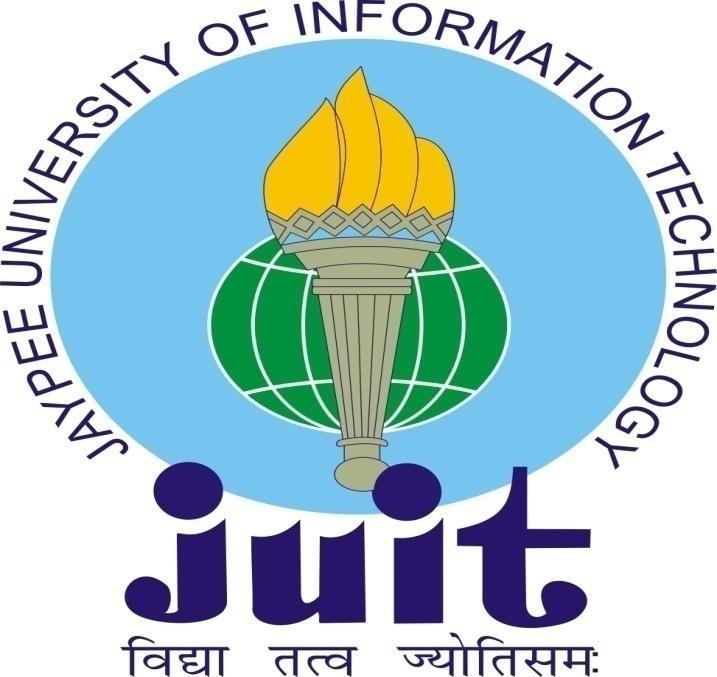
By

## YASH GOYAL(181216)

## SALONI SINGH(181246)

**UNDER THE SUPERVISION OF**

DR. HARI SINGH



Department of Computer Science & Engineering and Information Technology

# Jaypee University of Information Technology, Waknaghat, 173234, Himachal Pradesh, INDIA

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| **CONTENT** | **PAGE NO.** |
| **Declaration by candidate** | **I** |
| **Certificate by supervisor** | **II** |
| **Abstract** | **III** |
| **CHAPTER 1** | **Page 1** |
| **CHAPTER 2** | **Page 3** |
| **CHAPTER 3** | **Page 7** |
| **CHAPTER 4** | **Page 20** |
| **References** | **Page 24** |

**I**

**DECLARATION BY THE CANDIDATES**

We the undersigned solemnly declare that the project report

“TOPIC MODELLING USING MACHINE LEARNING AND PYSPARK” is based on our own work carried out during the course of our study under the supervision of Dr HARI SINGH .

We assert the statements made and conclusions drawn are an outcome of our own research work. We further certify that

I. The work contained in the report is original and has been done by us under the general supervision of our supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. We have followed the guidelines provided by the university in writing the report.

IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

YASH GOYAL(181216)

SALONI SINGH(181246)

**III**

**ABSTRACT**

Topic Modeling is an unsupervised learning technique which is used to extract hidden topics from a large volume of text data. We classify topics from unstructured data. It is a solution to a lot of real world problems like social media post topic determination, customer care text analysis etc. Therefore, it seems reasonable that given a community, we can cross-examine the textual content and the corresponding social relations obtained from external sources (such as Gmail) as complementary information to understand how one expresses language.

It answers questions like: For e.g In a customer care service centre when voice data is converted into text data so what exactly are they calling about? What exactly are they complaining about? So our aim is to find the exact topic for which the customer is calling about

There are several existing algorithms you can use to perform the topic modeling. The most common of it are, Latent Semantic Analysis (LSA/LSI), Probabilistic Latent Semantic Analysis (pLSA), and Latent Dirichlet Allocation (LDA)

With the prevalence of probabilistic topic modeling as a statistical approach to fit language, in this paper, we ask the question that, is it possible to consider social interaction effects jointly with statistical language model such that the resulting model incorporates social effects of the community group? We will provide intuition to the benefit of incorporating social information, and present a framework based on Latent Dirichlet Allocation for model and testing. A system is built to effectively crawl relevant information from heterogeneous online sources, and disambiguate entities that appear in multiple contexts. I test the proposed framework on a dataset consisting of over 10000 records within 20 categories (20\_news\_group). In this project we use PYSPARK as our dataset itself is a pretty big one so using pyspark will benefit us by faster implementation of the model. Experimental results show that incorporating social features does exhibit more explanatory power for language usage.

**PAGE 1**

**Chapter 01: INTRODUCTION**

**Introduction**

When one speaks, his or her choice of expression is often influenced by many salient factors: a news article she just read via iPhone; a thought from a discussion with a friend; the people involved in the current conversation, among others. There is a growing interest in particular to examine the effect of social interactions and relations to how one’s language pattern usage is affected. Previous studies have checked social structure, interaction patterns, incentives, to name a few. The increasingly available online social media services provide very good sources of information on social interactions. Services like Twitter and Facebook see usages by a wide range of users. Celebrities and professionals readily adapt to using such social media as a way of outreach to communities discussing issues of interest. In , tweets can be roughly categorized into four types: social stature, daily emotion, conversation, and information sharing. These usage purposes are tightly connected to social meanings. However, one of the major challenges in analyzing data is that many information sources that contain rich textual information often lack clear social context to derive social interaction-based analysis. For example, in online newspapers such as Boston Globe, very few articles directly specify multiple authors, although the article often involves multiple people in the process (editor, other colleagues writing stories in the same section, etc.) It is hard to directly derive social structure based on news stories alone. For this project, I aim to use TOPIC MODELLING to view social dimension and linguistic dimension separately to focus on the data preprocessing side of social signal incorporation, identification, and information transfer for prediction tasks. Ultimately, beyond the scope of this paper, is to deliver a perspective on a potential framework that can recover and automatically draw relations

**Objective of the Minor Project**

We will be using Topic modelling as a method for finding a group of words (i.e topic) from a collection of documents i.e our 20\_newsgroup dataset ,that best represents the information in the collection. It can also be thought of as a form of text mining – a way to obtain recurring patterns of words in textual material.

**Motivation of the Minor Project**

One of the most amazing things about topic modelling is that it has applications in a variety of areas. It can help motivate inquiry, provide unique insights into texts, and give you new ways to organize documents.

For e.g The Structural Topic Model (STM) is a form of topic modelling specifically designed with social science research in mind. STM allow us to incorporate metadata into our model and uncover how different documents might talk about the same underlying topic using different word choices.

**PAGE 2**

**Language Used**

Programming Language used is **PYTHON**

**Technical Requirements ( Hardware)**

As we will be working with Apache Spark(PYSPARK)

The minimum system requirements are

* Memory: 2 GB
* Graphics Card: NVIDIA GeForce GT 340
* CPU: Intel Core 2 Duo Q6867
* File Size: 50 MB
* OS: Windows XP

**Deliverables of the Minor Project**

The outcome of the project will be a final data frame corresponding to every category in our dataset consisting of a group of words which will decide in which category our input paragraphs falls into.

PAGE 3

**Chapter 02: MINOR PROJECT SDLC**

**Feasibility Study on Minor Project**

This project aims to explore the feasibility of using natural language processing and machine learning to automate topic extraction from a huge corpus of data to make it categorically easy to understand.

Qualitative studies, such as sociological research, opinion analysis and media studies, can benefit greatly from automated topic mining provided by topic models such as latent Dirichlet allocation (LDA).

**Requirements on Minor Project**

1. Functional Requirements

Machine Learning

* Programming language - Python
* Dataset - 20\_news\_group (10490 records)
* Natural Language Processing
* Libraries used for NLP - NLTK, SPARK NLP
* APACHE SPARK (PYSPARK)
* Data Preprocessing
* Data Analysis
* Term Frequency ( Count Vectoriser) , Inverse Document Frequency
* Stem function, Tokenizer,Lemmatization, Stopword
* Algorithm - Latent Dirichlet Allocation (LDA)

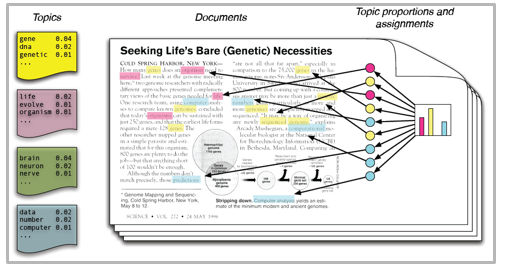
**2.** Non-Functional Requirements

ATLEAST:

* 2 GB memory
* a DUAL CORE CPU

**PAGE 4**

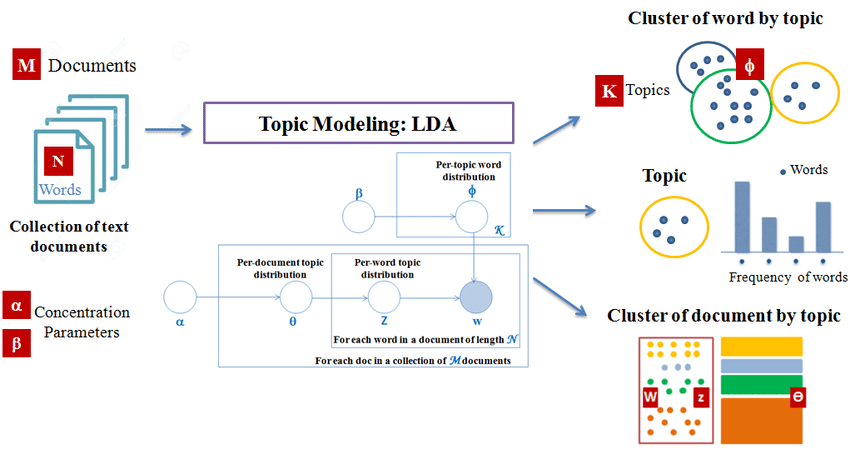
**Use Case Diagram of the Minor Project**

****

The Intuitions behind latent Dirichlet allocation. We assume that some number of 'topics’, which are distributions over words, exist for the whole collection (far left). Each document Is assumed to be generated as follows.

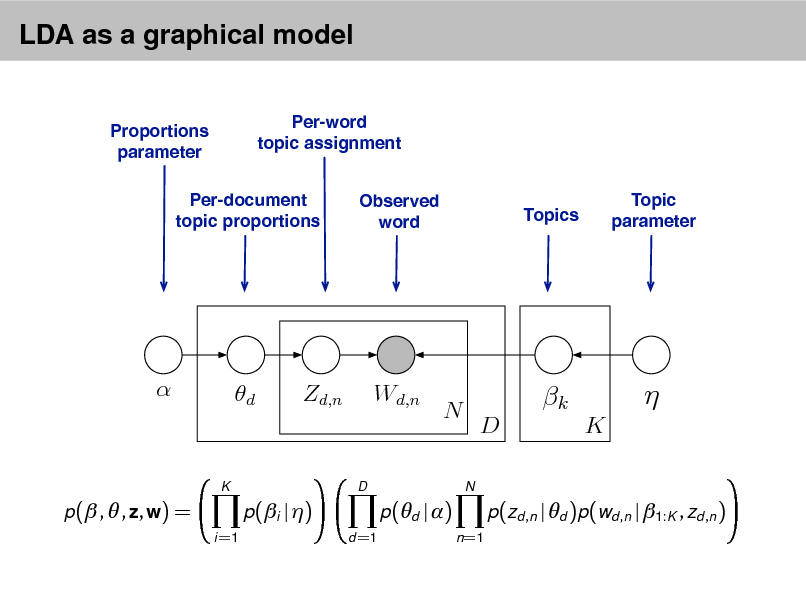
First choose a distribution over the topics (the histogram at right); then, for each word, choose a topic assignment (the colored calns) and choose the word from the corresponding topic.

**PAGE 5**

**DFD Diagram of the Minor Project**

* K is the number of topics
* N is the number of words in the document
* M is the number of documents to analyse
* 𝝰 is the Dirichlet-prior concentration parameter of the per-document topic distribution
* 𝝱 is the same parameter of the per-topic word distribution
* ф(k) is the word distribution for topic k
* 𝛉(i) is the topic distribution for document i
* z(i,j) is the topic assignment for w(i,j)
* w(ij) is the j-th word in the i-th document
* **ф** and 𝛉 are Dirichlet distributions, z and w are multinomials.

**PAGE 6**

**Transition Diagram of the Minor Project**

* This joint defines a posterior, **p(𝛉, z, 𝝱 | w)**.
* From a collection of documents, infer:

• Per-word topic assignment **Z**(d,n)

• Per-document topic proportions (d)

• Per-corpus topic distributions (k)

* Then use posterior expectations to perform the task at hand: information retrieval, document similarity, exploration, and others.

**PAGE 7**

**Chapter 03: IMPLEMENTATION OF THE MINOR PROJECT**

**3.1 DataSet Used in the Minor Project**

**Dataset Name -** 20Newsgroup

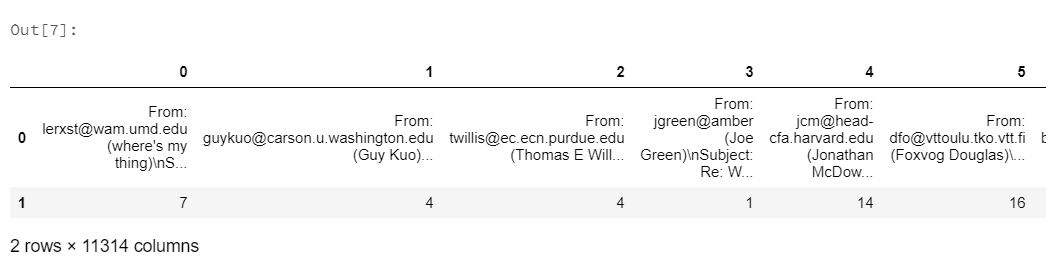
**Dataset link -**[*http://qwone.com/~jason/20Newsgroups/*](http://qwone.com/~jason/20Newsgroups/)

FIG 1 20Newsgroup DATASET (RAW)

**3.2 Date Set Features**

**3.2.1 Types of Data Set**

The dataset is available in .tar.gz bundles. Each subdirectory in the bundle represents a newsgroup; each file in a subdirectory is the text of some newsgroup document that was posted to that newsgroup.

**3.2.2 Number of Attributes, fields, description of the data set**

The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups.This dataset has approximately 11314 entries in form of columns. To the best of my knowledge, it was originally collected by Ken Lang, probably for his [*Newsweeder: Learning to filter netnews*](http://qwone.com/~jason/20Newsgroups/lang95.bib) paper, though he does not explicitly mention this collection. The 20 newsgroups collection has become a popular data set for experiments in text applications of machine learning techniques, such as text classification and text clustering.

The data is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other (e.g. comp.sys.ibm.pc.hardware / comp.sys.mac.hardware), while others are highly unrelated (e.g misc.forsale / soc.religion.christian). Here is a list of the 20 newsgroups, partitioned (more or less) according to subject matter:

**PAGE 8**

Table 1 - Dataset 20 Newsgroup categories.

|  |  |  |
| --- | --- | --- |
| comp.graphics  comp.os.ms-windows.misc  comp.sys.ibm.pc.hardware  comp.sys.mac.hardware  comp.windows.x | rec.autos  rec.motorcycles  rec.sport.baseball  rec.sport.hockey | sci.crypt  sci.electronics  sci.med  sci.space |
| misc.forsale | talk.politics.misc  talk.politics.guns  talk.politics.mideast | talk.religion.misc  alt.atheism  soc.religion.christian |

**3.3 Design of Problem Statement**

**With the growth of online social network platforms and applications, large amounts of textual user-generated content are created daily in the form of comments, reviews, and short-text messages. As a result, users often find it challenging to discover useful information or more on the topic being discussed from such content. Machine learning and natural language processing algorithms are used to analyze the massive amount of textual social media data available online, including topic modeling techniques that have gained popularity in recent years.**

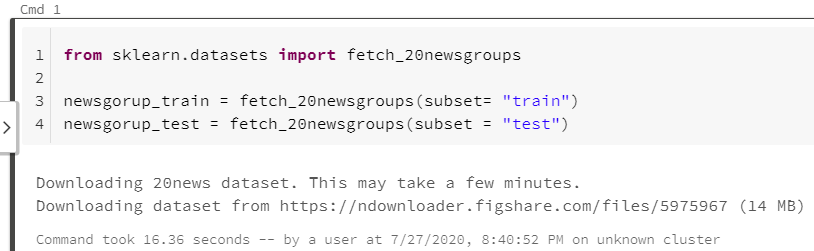
**From the given raw dataset, perform everything needed from text cleaning to building a ML model using TOPIC MODELLING and extract useful topics from all the entries in the dataset and assign them to categories they belong to to check model accuracy, as it is a NLP based model you will need to manually check if the model is working right or not.**

**PAGE 9**

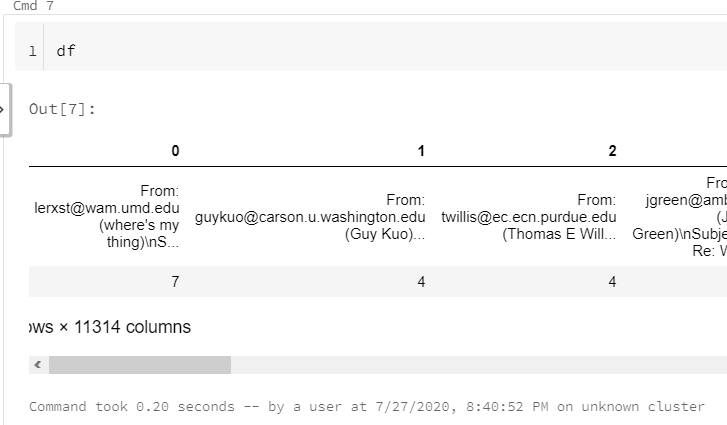
**3.4 Algorithm / Pseudo code of the Project Problem with the screenshots of various stages of the project**

1. DATASET

This is our dataset (20\_news group). It has 11314 columns in the form of email like text. This text data has a lot of noise in it and is completely unstructured. We perform data cleaning and data preprocessing on it . We convert this dataset into a spark dataframe and perform basic text cleaning to it, but it still has a lot of noise so we perform deep data preprocessing.

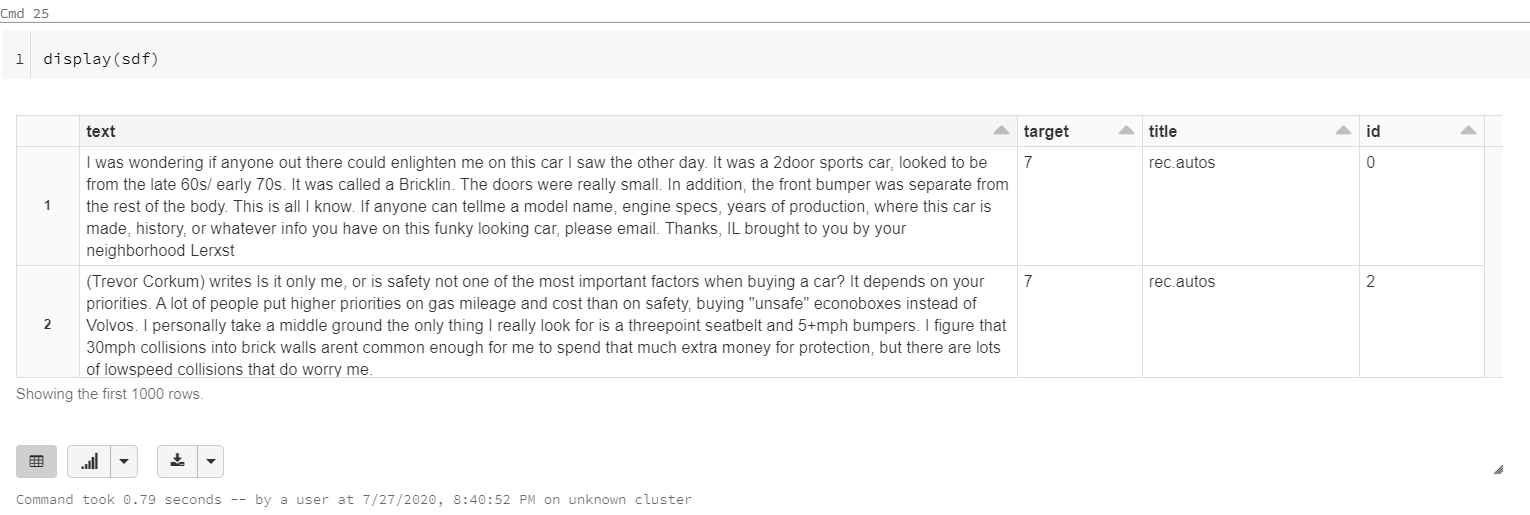
****

**FIG - DOWNLOADING THE DATASET**

****

**FIG - OUR DATASET**

**PAGE 10**

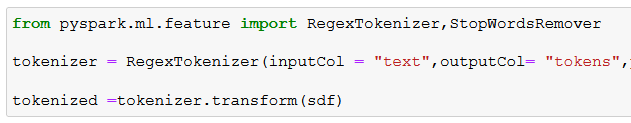
****

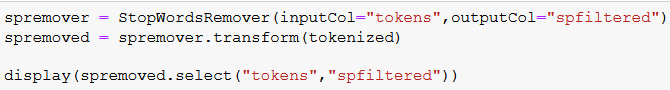
**FIG- DATASET AFTER CLEANING**

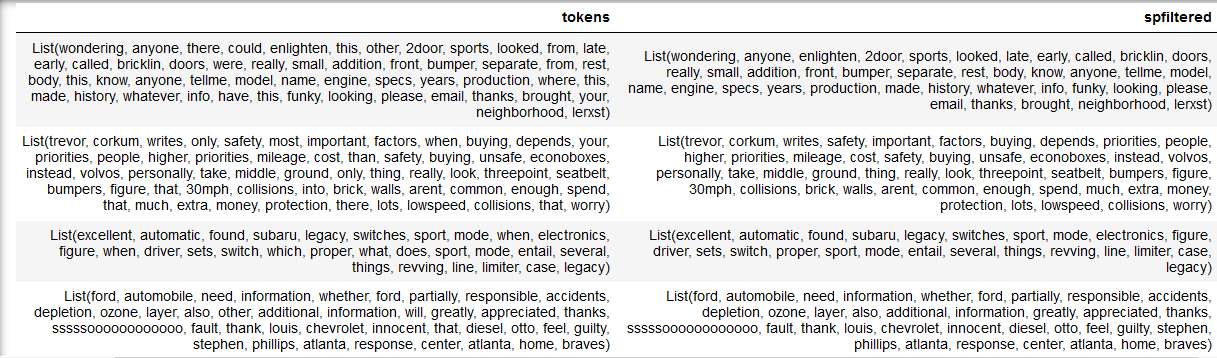
1. STARTING ON THE NLP PIPELINE

REGEX TOKENIZER - It tokenizes the text paragraphs into a list of individual words which gives us easy access to all the vocab words in the text.

STOP WORD REMOVAL - Stopwords are the common words which are found almost in every sentence for e.g this,that,is,a ,and etc. The data was not without noise after this so we had to create our own stop words list and remove those words from spfiltered.

****

****

****

**PAGE 11**

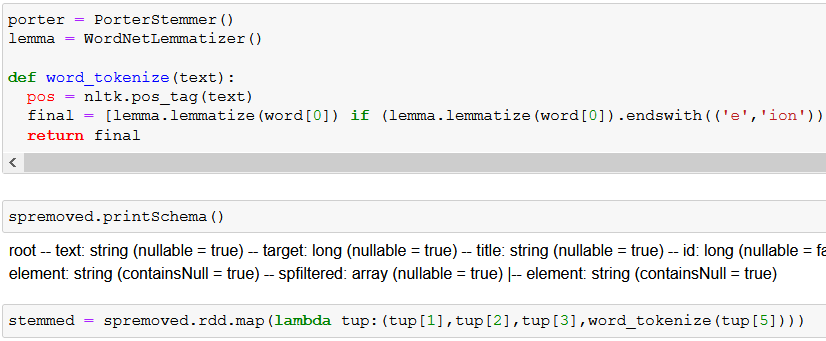
1. Stemming and Lemmatization - not in pyspark so we used nltk

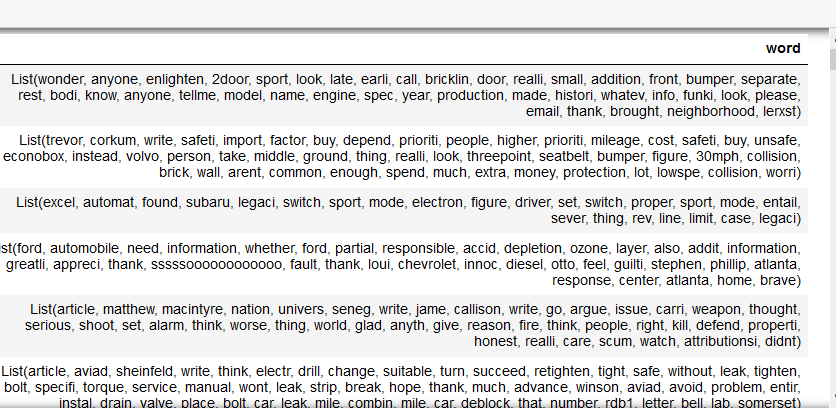
Stemming - it when performed generates the root form of every word in the list. For e.g succeed, successful turns to success.

Lemmatization - it truly understands the dataset and gives better output then stemming for e.g

a lemmatization algorithm would know that the word better is derived from the word good, and hence, the lemma is good. But a stemming algorithm wouldn’t be able to do the same. There could be over-stemming or under-stemming, and the word better could be reduced to either bet, or bett, or just retained as better. But there is no way in stemming that it could be reduced to its root word good. This, basically is the difference between stemming and lemmatization.

WE DO THIS SO THAT WHEN WE WILL BE COUNTING THE WORD FREQUENCY WE WILL NOT BE DOUBLE COUNTING ANYTHING

****

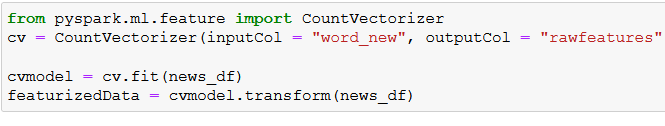
****

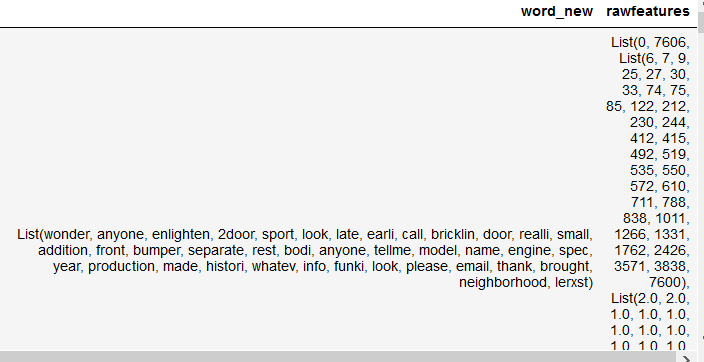
1. CALCULATING TERM FREQUENCY

It means how often a term occurs in a document, in context of NLP terms corresponds to words or phrases. We use sparks COUNT VECTORIZER to do this.

**PAGE 12**

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

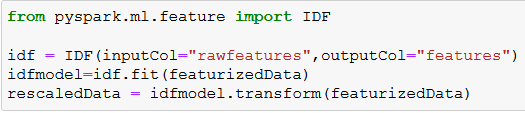
****

****

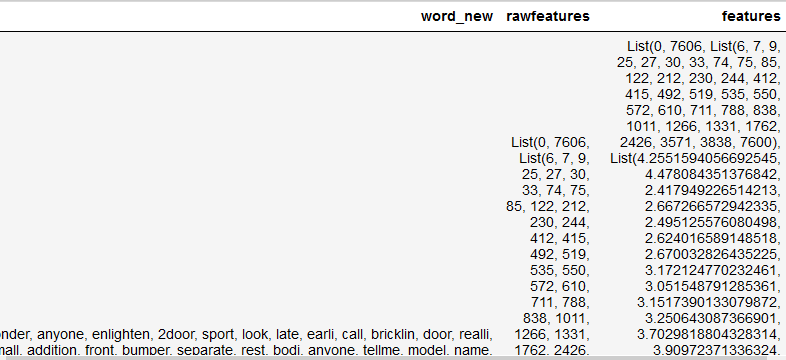
1. CALCULATING INVERSE DOCUMENT FREQUENCY

The inverse document frequency of the word across a set of documents. This means, how common or rare a word is in the entire document set. The closer it is to 0, the more common a word is. This metric can be calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

****

****

**PAGE 13**

****

TF-IDF is a statistical measure that evaluates how relevant a word is to a document in a collection of documents. This is done by multiplying two metrics: how many times a word appears in a document, and the inverse document frequency of the word across a set of documents

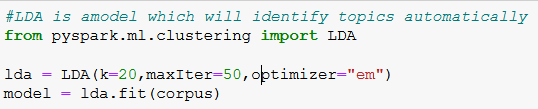
After this we calculate TF\*IDF i.e (rawfeatures\*features) and feed it to the LDA algorithm

1. LDA ALGORITHM - LATENT DIRICHLET ALLOCATION Latent: This refers to everything that we don’t know a priori and are hidden in the data. Here, the themes or topics that document consists of are unknown, but they are believed to be present as the text is generated based on those topics. Dirichlet: It is a ‘distribution of distributions’. Suppose there is a machine that produces dice and we can control whether the machine will always produce a dice with equal weight to all sides, or will there be any bias for some sides. So, the machine producing dice is a distribution as it is producing dice of different types. Also, we know that the dice itself is a distribution as we get multiple values when we roll a dice. This is what it means to be a distribution of distributions and this is what Dirichlet is. Here, in the context of topic modeling, the Dirichlet is the distribution of topics in documents and distribution of words in the topic. It might not be very clear at this point of time, but it’s fine as we will look at it in more detail in a while. Allocation: This means that once we have Dirichlet, we will allocate topics to the documents and words of the document to topics.

**We cache the id and features column because they include the performance of the model, When we cache them in the memory we speed up our model performance**

****

**PAGE 14**

****

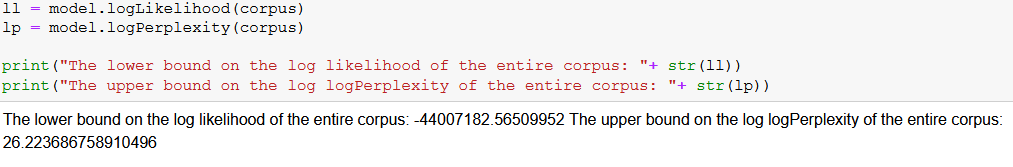
1. Output methods to evaluate the model

#these are indicative measure

#While comparing two different models we use the below methods

#for different iteration of LDA then we can use these

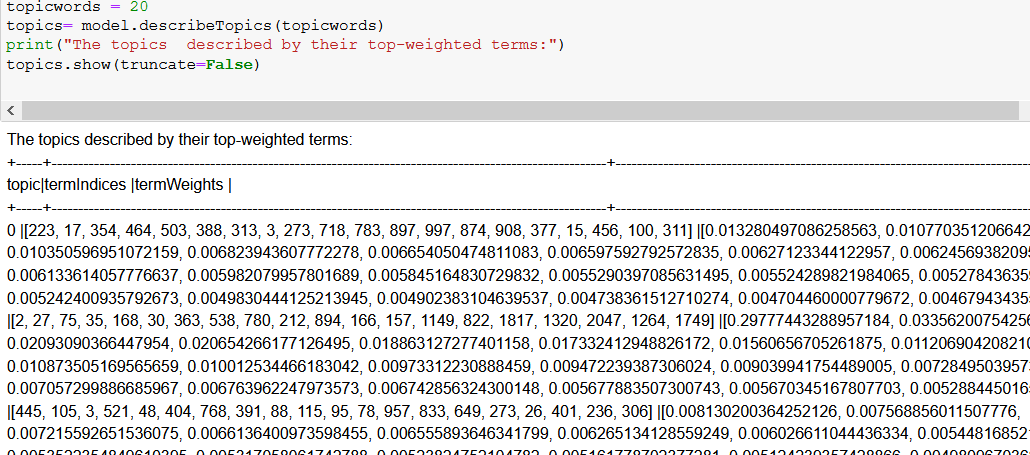
#The lesser the score the better

****

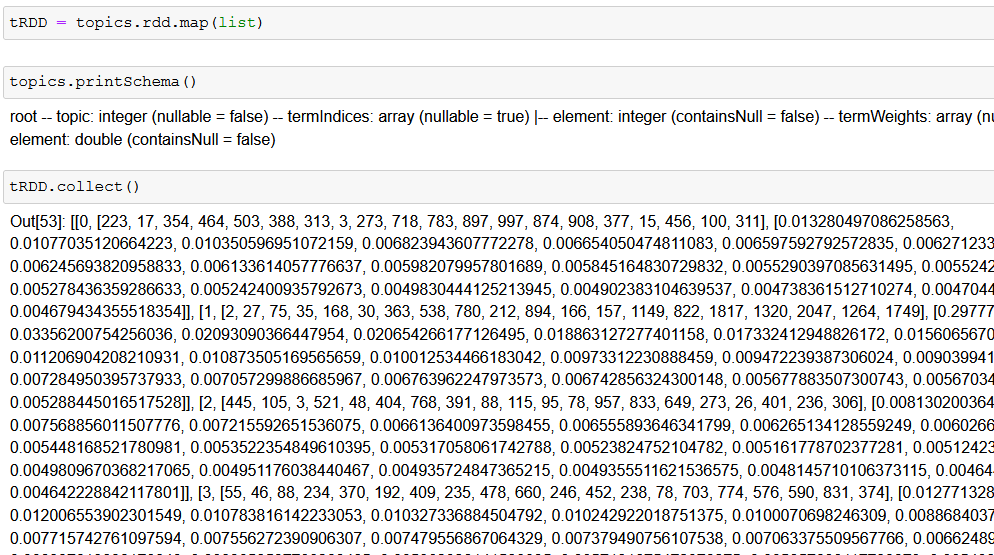
1. The only correct way to check the Accuracy of Topic Modelling is to classify our own test data and check it against the model output i .e the best way to evaluate a topic model is we classify our test data for e.g like what can be the possible topic and then we see manually if the model is predicting properly.

**PAGE 15**

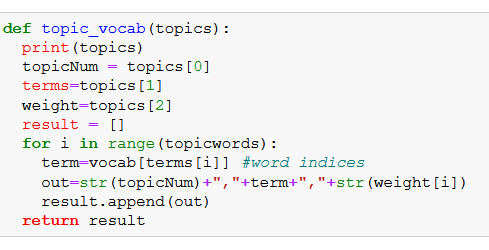
1. Here For e.g we are taking 20 inputs and we are extracting the top 20 topic words from them based on their topic indices.

****

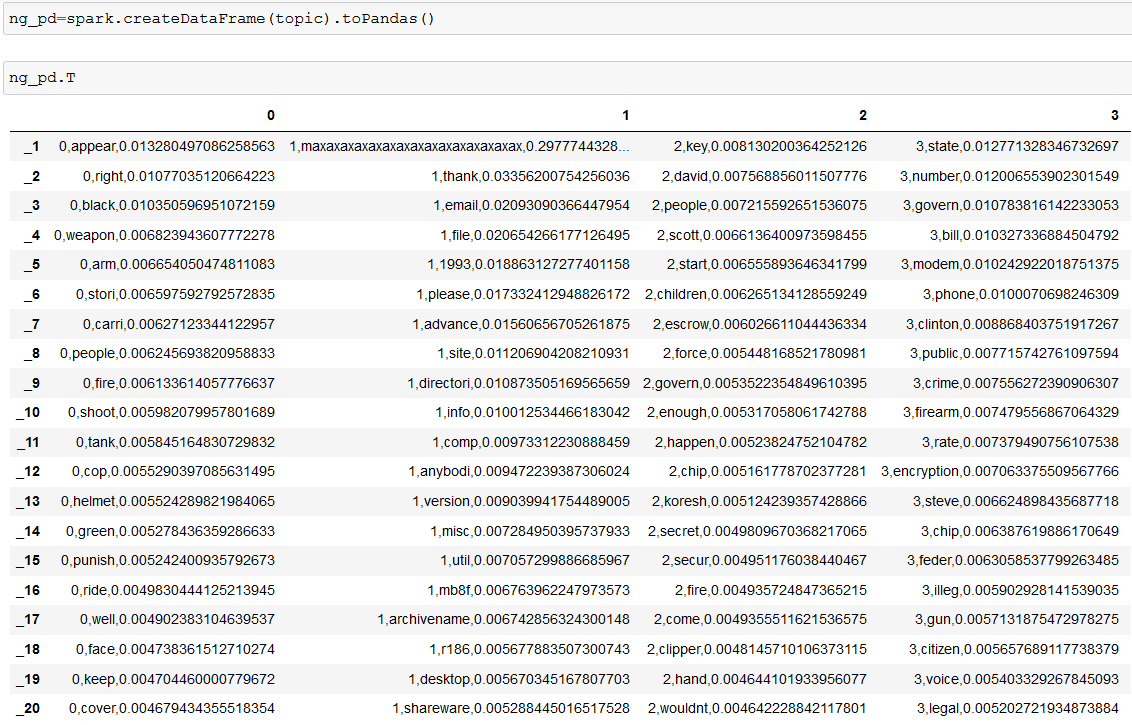
1. Now as we have the word in the form of topic indices what we are going to do is we are going to feed those scores to the python vocabulary by converting topic variable into a RDD function.

****

**PAGE 16**



1. FINALLY WE CONVERT THE SPARKS DATAFRAME TO PANDAS AND OUTPUT IT

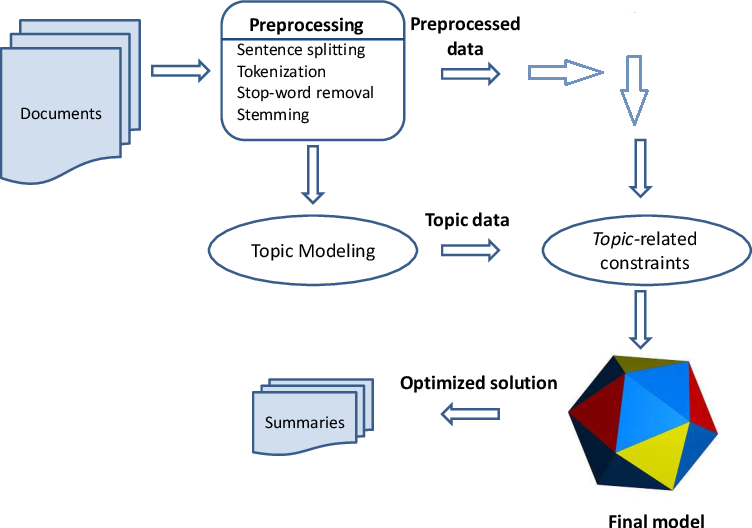
****

**PAGE 17**

****

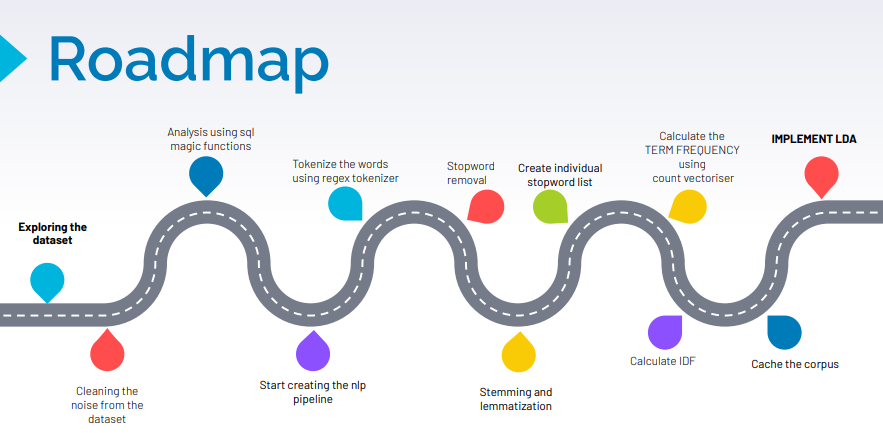
**PAGE 18**

**3.5 Flow graph of the Minor Project Problem**

****

**FIG - FLOW GRAPH**

**PAGE 19**

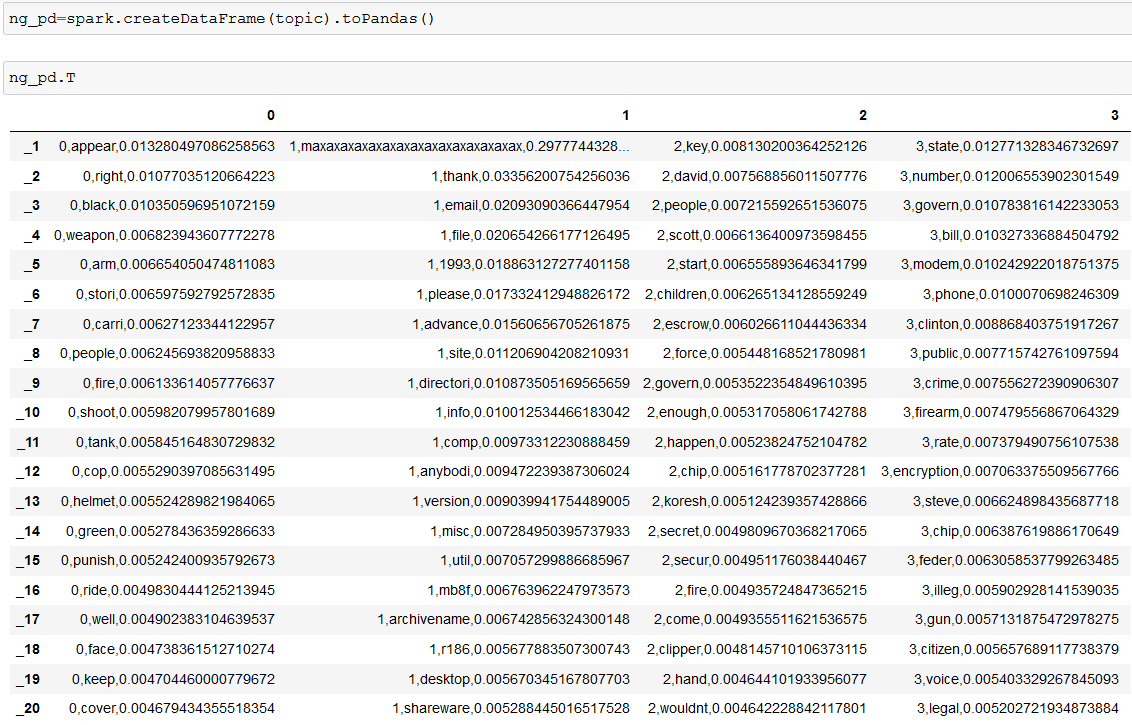
****

**FIG - ROADMAP OF MINOR PROJECT**

**PAGE 20**

**Chapter 04: RESULTS**

**4.1 Discussion on the Results Achieved**

****

****

PAGE 21

Table 1 - Dataset 20 Newsgroup categories.

|  |  |  |
| --- | --- | --- |
| comp.graphics  comp.os.ms-windows.misc  comp.sys.ibm.pc.hardware  comp.sys.mac.hardware  comp.windows.x | rec.autos  rec.motorcycles  rec.sport.baseball  rec.sport.hockey | sci.crypt  sci.electronics  sci.med  sci.space |
| misc.forsale | talk.politics.misc  talk.politics.guns  talk.politics.mideast | talk.religion.misc  alt.atheism  soc.religion.christian |

The achieved results were upto the mark as we ran our code for 20 inputs with 50 iterations along 20 categories and we evaluated the model manually classifying the extracted topic words to the categories which are present in our dataset. ( THIS HELPS THE COMPANY TO KNOW WHAT EXACTLY IS SOMEONE TALKING ABOUT IN A PARTICULAR CONVERSATION)

For example

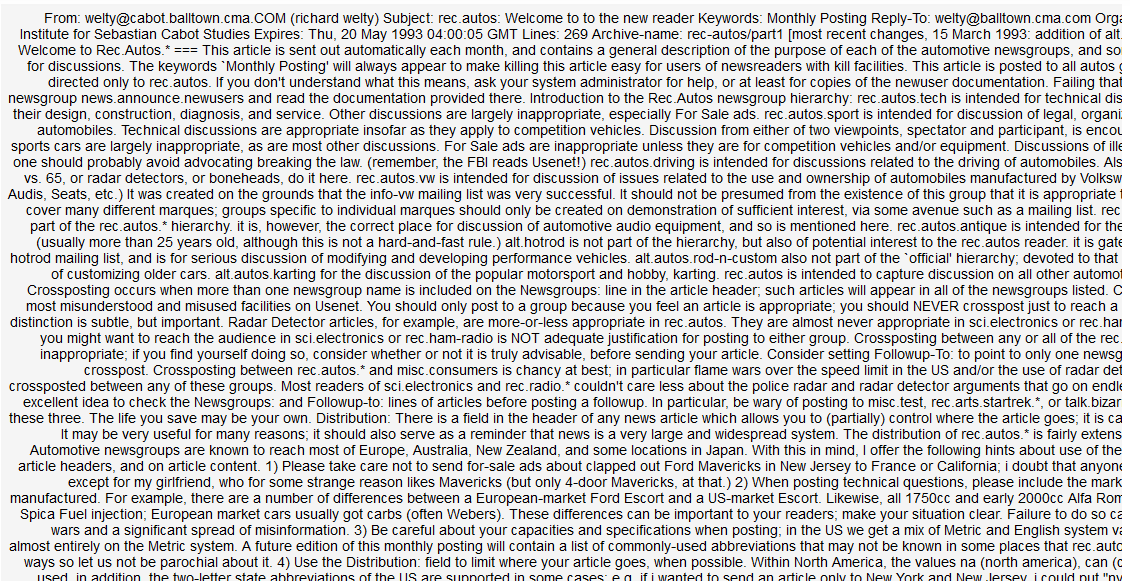


FIG - ‘1/11213 ENTRIES WE HAVE IN THE DATASET’

This is the kind of entries we have in the dataset, what our model is doing is it is extracting whatever topic word it thinks is relevant according to the instructions we gave it in the code. Using those topic words companies can classify a certain 3000 word email into a category and programme a common auto reply for it which saves a lot of time, effort and resources.

PAGE 21

**PAGE 22**

**4.2 Applications of the Minor Project**

* Text summarisation. We live in the era of unprecedented amounts of data available online. Text summarisation can help us summarise key points needed for a particular application, business or research objective, and topic modelling can improve the performance of such models, yielding better summaries of said content.
* Query expansion. As topic models uncover relationships among words through latent topics, they can exploit the relationships in order to expand shorter queries on the semantic level, which can significantly improve the performance of search engines.
* Sentiment analysis. Sentiment analysis deals with the extraction of sentiments and opinions of various groups of people (customers, stakeholders, investors, etc.). A big challenge there is deriving a useful numerical variable from text, and topic models can help. As one example, enriching such models with topic-term distribution could help airlines better categorise passengers’ online reviews, improving the provided services in key points for maximum effect on revenues.
* Recommender systems. Similar to sentiment analysis, topic models in recommender systems can better group various services or products in comparison to more traditional clustering algorithms, ultimately resulting in more appropriate matching of users and products.
* Blockchain. Though cryptocurrencies are growing in popularity, they remain risky to use due to their volatile and unregulated market. Topic modelling can help assess large quantities of unstructured information available online from Bitcoin developers and investors, improving the automatic detection of fraudulent activities, risk levels, and even future events on the market.
* Understanding scientific publications. As with data, the amount of knowledge generated today remains unmatched in both breadth and extent, growing at greater speeds than has been noted in the modern world. It is becoming increasingly difficult to promptly and efficiently find needed information and assess the reliability and value of that information for research or business needs. More so, there is a growing need to systematise all generated knowledge.
* And many more usages …

**PAGE 23**

**4.3 Limitations of the Minor Project**

(1) The number of documents plays perhaps the most important role; it is theoretically impossible to guarantee identification of topics from a small number of documents, no matter how long. Once there are sufficiently many documents, further increasing the number may not significantly improve the performance, unless the document length is also suitably increased. In practice, the LDA achieves comparable results even if thousands of documents are sampled from a much larger collection.

(2) The length of documents also plays a crucial role: poor performance of the LDA is expected when documents are too short, even if there is a very large number of them. Ideally, the documents need to be sufficiently long, but need not be too long: in practice, for very long documents, one can sample a fraction of each document and the LDA still yields comparable topics.

(3) When a very large number of topics then needed are used to fit the LDA, the statistical inference may become inescapably inefficient. In theory, the convergence rate deteriorates quickly to a nonparametric rate, depending on the number of topics used to fit the LDA. This implies, in practice, the user needs to exercise extra caution to avoid selecting an overly large number of topics for the model.

**4.4 Future Work**

PAGE 24

**References**

1. *Li, S. (2018, March 30). Topic Modelling in Python with NLTK and Gensim [Article]. Retrieved from* [*https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21*](https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21)
2. *Alam, K. M., Hemel, M. T. H., Muhaiminul Islam, S., & Akther, A. (2020, December). Bangla News Trend Observation using LDA Based Topic Modeling. 2020 23rd International Conference on Computer and Information Technology (ICCIT).* [*https://doi.org/10.1109/iccit51783.2020.9392719*](https://doi.org/10.1109/iccit51783.2020.9392719)
3. *Alzghool, M., & Inkpen, D. (2008). Clustering the topics using TF-IDF for model fusion. Proceeding of the 2nd PhD Workshop on Information and Knowledge Management - PIKM ’08.* [*https://doi.org/10.1145/1458550.1458569*](https://doi.org/10.1145/1458550.1458569)
4. *Dai, W. (2018). Improvement and Implementation of Feature Weighting Algorithm TF-IDF in Text Classification. Proceedings of the 2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018).*
5. [*https://doi.org/10.2991/ncce-18.2018.94*](https://doi.org/10.2991/ncce-18.2018.94)
6. *JING, Y., GOU, H., & SUN, W. (2019). Text Classification Based on LDA and Semantic Analysis. DEStech Transactions on Computer Science and Engineering, iccis.* [*https://doi.org/10.12783/dtcse/iccis2019/31986*](https://doi.org/10.12783/dtcse/iccis2019/31986)
7. *L. Jockers, M., & Thalken, R. (2020). Topic Modeling. Text Analysis with R, 211–235.* [*https://doi.org/10.1007/978-3-030-39643-5\_17*](https://doi.org/10.1007/978-3-030-39643-5_17)
8. *Nam, S., & Lee, H. C. (2019). Airline Passenger Characterizations Using LDA Topic Modeling. KOREAN MANAGEMENT SCIENCE REVIEW, 36(3), 67–85.* [*https://doi.org/10.7737/kmsr.2019.36.3.067*](https://doi.org/10.7737/kmsr.2019.36.3.067)
9. *Tong, Z., & Zhang, H. (2016, May). A Text Mining Research Based on LDA Topic Modelling. Computer Science & Information Technology ( CS & IT ).*

[*https://doi.org/10.5121/csit.2016.60616*](https://doi.org/10.5121/csit.2016.60616)