Recommendation System using Latent Dirichlet Allocation

Vipul Sangode

Recommendation System using Latent Dirichlet Allocation

Thesis submitted to in partial fulfillment
of the requirements for the degree

of

Bachelor of Technology

in

Computer Science and Engineering (CSE)

by

Vipul Sangode

Under the guidance of

Dr. Ramakrishna Bandi

Assistant Professor (Mathematics)

IIIT Naya Raipur

Dr. SPM International Institute of Information Technology
Naya Raipur, India 493661

May 2023

This work is dedicated to my parents, Family and Friends for their Support, Sacrifice Encouragement and love

Approval of the Viva-Voce Board

May 16, 2023

Certified that the thesis entitled **Recommendation System using Latent Dirichlet Allocation** submitted by **Vipul Sangode** to the Dr. SPM International Institute of Information Technology, Naya Raipur, India, for the award of the degree of Bachelor of technology has been accepted by the examiners and that the student has successfully defended the thesis in the viva-voce examination held today.

Dr. Ramakrishna Bandi
Assistant Professor, Mathematics,
IIIT-Naya Raipur.

Dr. Vivek Tiwari
Assistant Professor, CSE, IIIT-Naya
Raipur.

Srinivasa K G
Professor, Dean(R&I), HOD DSAI,
IIIT-Naya Raipur.

DECLARATION

May 16, 2023

I certify that

a. The work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Vipul Sangode

Date: May 16, 2023

Certificate

This is to certify that the thesis entitled, **Recommendation System using Latent Dirichlet Allocation** submitted by **Vipul Sangode** to Dr. SPM International Institute of Information Technology, Naya Raipur, Chhattisgarh, India, is a record of bona fide research work under my supervision and I consider it worthy of consideration for the award of the degree of Bachelor of Technology of the Institute.

Signature of the Supervisor: ....

Name of the Supervisor: **Dr. Ramakrishna Bandi**,

Designation: Assistant Professor
Institute: IIIT Naya Raipur

Acknowledgment

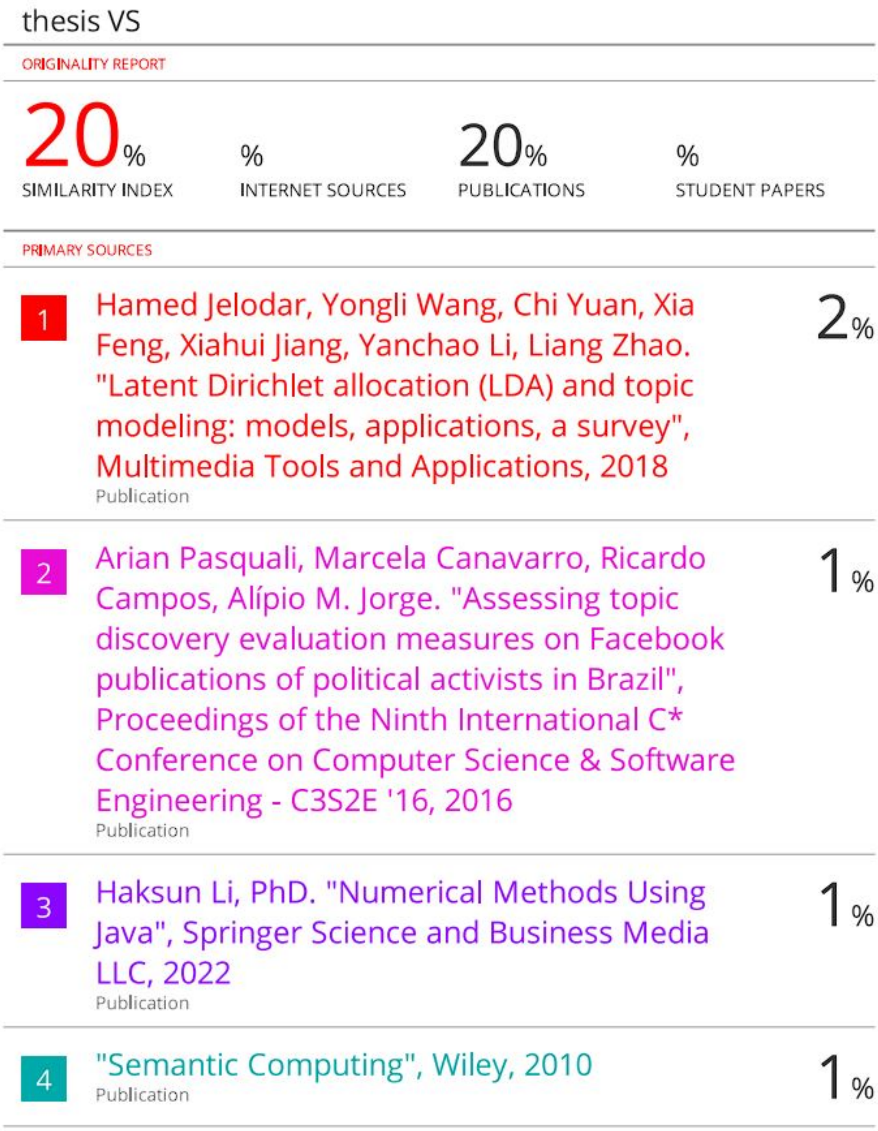
To **Dr. Ramakrishna Bandi**, Assistant Professor(Mathematics), I would like to express my profound gratitude for his essential direction, support, and encouragement throughout this endeavour. He kindly offered his knowledge and provided me helpful criticism that significantly raised the calibre of this job. Additionally, I would like to express my gratitude to the professors from Computer Science and Engineering department for their insightful comments and recommendations that influenced the course of this project. Their opinions and suggestions made it easier for me to comprehend the topic at hand. I want to express my gratitude to my friends and coworkers for their moral support and encouragement throughout this effort. I was able to maintain my motivation and focus thanks to their words of support and advice.

In closing, I want to express my gratitude to my family for their consistent support, inspiration, and tolerance during my academic career. They have always been a source of strength for me with their love and support. I would like to convey my appreciation to all of my teachers who have shared their expertise with me to help me become who I am. I will never be able to express my gratitude to my parents, brother, and other family members enough for their inspiration and assistance.

May 16, 2023
Naya Raipur

Vipul Sangode

Plagiarism Report



thesis VS

ORIGINALITY REPORT

20%	%	20%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Hamed Jelodar, Yongli Wang, Chi Yuan, Xia Feng, Xiahui Jiang, Yanchao Li, Liang Zhao. "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey", Multimedia Tools and Applications, 2018 Publication	2%
2	Arian Pasquali, Marcela Canavarro, Ricardo Campos, Alípio M. Jorge. "Assessing topic discovery evaluation measures on Facebook publications of political activists in Brazil", Proceedings of the Ninth International C* Conference on Computer Science & Software Engineering - C3S2E '16, 2016 Publication	1%
3	Haksun Li, PhD. "Numerical Methods Using Java", Springer Science and Business Media LLC, 2022 Publication	1%
4	"Semantic Computing", Wiley, 2010 Publication	1%

Abstract

Natural Language Processing (NLP) is becoming increasingly important in today's world because it allows machines to understand, interpret, and interact with human language in a way that was previously impossible. NLP has the potential to transform the way we interact with technology, enabling more intuitive and effective communication and improving many aspects of our lives

The quantity of entertainment alternatives available to Netflix subscribers is obscene, with over 6000 films and TV shows in a sample dataset. As a result, the goal of this project is to provide a recommendation for a TV show or movie using a straightforward content-based recommendation system. Topic Modelling is an unsupervised machine learning technique in which documents are grouped according to how closely their contents match. Latent Dirichlet Allocation is a well-liked algorithm. Each topic in LDA represents a probability distribution of words, and each document represents a probability distribution of topics. In the multi-dimensional vector space, clusters are formed as a result of the documents' increasing similarity with one another.

In this article, I have explored Latent Dirichlet Allocation (LDA), a generative probabilistic model for discrete data collections like text corpora. Each item of a collection is modelled as a finite mixture over an underlying set of themes/topics in the three-level hierarchical Bayesian LDA model. The model for each topic is an infinite mixture over a base set of topic probabilities. The topic probabilities in the context of text modelling offer an explicit representation of a document.

Keywords: Distribution, Latent Dirichlet Allocation, NLP, Recommendation, Topic Modelling

Contents

List of Abbreviations

LDA: Latent Dirichlet Allocation

NLP: Natural Language Processing

IR: Information Retrieval

TF-IDF: Term Frequency Inverse Document Frequency

MBD: Multivariate Beta Distribution

List of Figures

CHAPTER 1

Introduction

Preface

This chapter presents the introduction of the thesis containing brief description of how topic modelling is used as an unsupervised machine learning technique for the purpose of mining topic evolution. The study suggests a model based on Latent Dirichlet Allocation (LDA). In order to determine the trend of topic intensity, we start by collecting all of the reviews, using LDA to identify themes and their key phrases. We next obtain the probability distribution of document - topic on various time periods. Finally this is used to create a content-based recommendation system .

1.1 Introduction

In this paper we consider the problem of modeling text corpora and other collections of discrete data. The goal is to extract latent topics or themes of the members of a collection that enable efficient processing of large corpus while preserving the essential statistical relationships that are useful for basic tasks such as classification, novelty detection, text summarization, and similarity and relevance judgments.

With time, data is expanding dramatically. The data is largely unstructured, and some of it isn't even labelled, for example, customer reviews on an amazon product, Elon Musk tweets on Ukraine-Russia War, or just a bunch of college student assignments with mere headings as "Assignment Number X". It is extremely time-consuming to manually label each and every piece of data as we cannot possibly predict what content can the following data hold. How else can we label such a large amount of data if not manually? The LDA will come to our aid. LDA is one topic modelling method used to look at a lot of data, organise the data into related groups, and label each group. As mentioned above, LDA is used as unsupervised machine learning technique which uses probability distributions to group data as per related topics. These topics may not necessarily mean anything to the machine except that the machine finds relations between word-topic and topic-document and groups the data into different topics. In other words, the machine may group the data into 20 different topics or whatsoever number of topics we want to divide the data into and the machine will give topics as "Topic-1", "Topic-2", and so on. It is the user or customer's duty to identify the words in the specific topics and give the topics relevant names. This makes the topic modelling task quite subjective to the user's or customer's end implementation or requirement. "Physics" and "Chemistry" might have quite different meanings and background for a teacher or a science student as a topic, but for a politician customer, these both topics might be vaguely put into a single topic as "Science". An overview of LDA model is given is figure 1.1

LDA was developed in 2003 by researchers Michael Jordan, Andrew Ng, and David Blei. As a result of its effectiveness, simplicity, and appeal to the intuitive, it has received strong support for use. Lets see what each term in the title has to hold:

- **Latent:** Latent refers to everything that is hidden in the data and that we do not know a priori. In this case, the themes or topics that make up the document are unknown, but they are assumed to be there because the text was produced using those themes.

- **Dirichlet:** It is called as 'distribution of distributions' or even sometimes as a 'family of distributions'. When modelling multivariate categorical data, such as the distribution of

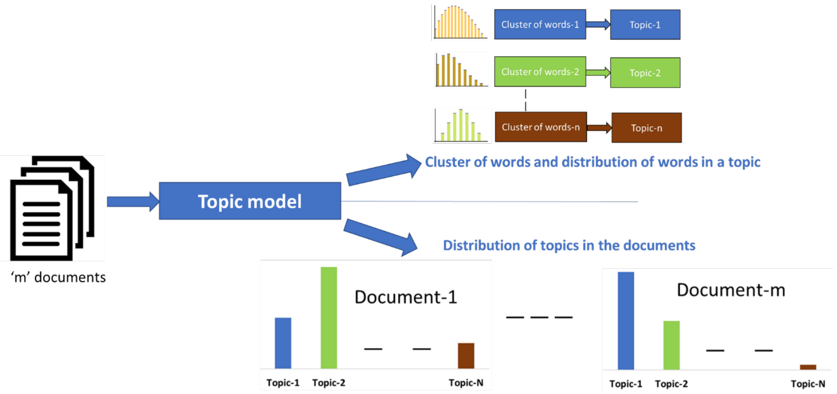


Figure 1.1: LDA: Latent Dirichlet Allocation model overview

word frequencies in a document, a continuous probability distribution is frequently utilised called the Dirichlet distribution. A group of K positive real parameters, represented by α1, α2,..., αK, that control the distribution's form define the Dirichlet distribution. These variables represent preconceived notions about how the category variables are distributed. But what does this mean as 'distribution of distributions'? Consider a device which produces dice. We can decide whether the device will always manufacture a die with equal weight on all sides or whether there will be bias for some sides.

As a result, since it generates dice of various varieties, this machine creating dice is considered a distributor. Since we can get numerous numbers when we roll a die, whether fairly or unfairly, we also know that the dice themselves are a distribution. A distribution of distributions is exactly what Dirichlet is, and this is what it means to be a distribution of distributions. It will be discussed in detail in later sections.

- **Allocation:** Once we have Dirichlet distribution with us, we will allocate topics to the documents and words of the document to topics.

1.2 Review of Literature

Topic Modelling is a type of tagging used in unsupervised machine learning tasks, primarily for information retrieval where it aids with query extension. It is widely used by search developers to map user preferences in topics. Classification, categorization, and document summarization are the basic uses of topic modelling. Topic Modelling is linked to AI approaches for genomics, social media, and computer vision problems. Additionally, it enables analysis of user sentiment on social networks. This approach can be used to automatically identify social circles among a subject's friends. We evaluate both the profile features and users' friends for each friend, which we shall refer to as documents; these are both referred to as ”tokens.” Topic modelling is distinct from text classification and clustering tasks because it uses unstructured data. Topic Modelling does not seek to identify commonalities between texts, unlike text classification or clustering, which tries to simplify information retrieval and create clusters of documents. Typically, there are several themes and a variety of texts in topic modelling.

Latent Dirichlet Allocation, Latent Semantic Analysis, Correlated Topic Modelling, and Probabilistic Latent Semantic Analysis are a few of the methods used for topic modelling jobs.

Algorithms other than LDA:

- **Latent Semantic Analysis:** This algorithm assists in maintaining texts and words in a semantic space for classification using a technique called singular value decomposition.

- **Probabilistic Latent Semantic Analysis:** PLSA, or Probabilistic Latent Semantic Analysis, uses the likelihood that a word is related to a topic and a topic in a document and can be trained using an expectation-maximization algorithm. The multinomial distribution of words is the foundation of this methodology.

- **Correlated Topic Modelling:** A more thorough comprehension of the connections between topics is made possible by the Correlated Topic Model (CTM), a hierarchical model that explicitly models the correlation of latent topics.

LDA, or Latent Dirichlet Allocation, is the best and most widely used algorithm to define and work with topic modelling. It extracts topic probabilities from available statistical data. LDA is a succesor of probabilistic latent semantic indexing.

In this paper, we use the terminology of text collections to refer to things like ”words,” ”documents,” and ”corpora.” This is beneficial because it helps in directing intuition, especially when we introduce latent variables designed to capture abstract concepts like topics. It is crucial. It is important to keep in mind, however, that the LDA model is not confined to text and can be used to solve other issues involving data collections, including information from fields like collaborative filtering, content-based image retrieval, and bioinformatics.

Researchers in the field of **information retrieval** (IR) have made significant advancements in solving former mentioned issue (Baeza-Yates and Ribeiro-Neto, 1999). The fundamental method for text corpora proposed by IR researchers reduces each document in the corpus to a vector of real numbers, each of which represents ratios of counts. This method is successfully used in modern Internet search engines. According to the well-known **tf-idf** scheme (Salton and McGill, 1983), a basic vocabulary of ”words” or ”terms” is selected, and a count of each word's occurrences is created for each document in the corpus. This term frequency count is normalised appropriately before being compared to an inverse document frequency count, which counts the occurrences of a word across the entire corpus (typically on a log scale, and again, appropriately normalised). A term-by-document matrix X with columns containing the tf-idf values for each document in the corpus is the final output. As a result, documents of any length can be converted into fixed-length lists of numbers using the tf-idf scheme.

$$TF(t,d) = \frac{\text{number of occurences of term 't' in document 'd'}}{\text{Total number of terms in the document 'd'}} \tag{1.1}$$

$$IDF(t) = \log(\frac{\text{total number of documents in corpus}}{\text{Number of documents with term 't' in them}}) \tag{1.2}$$

Even though the tf-idf reduction has some appealing characteristics, most notably in its quick identification of word sets that are discriminative for documents in the collection, the method only offers a modest reduction in description length and reveals limited in the way of statistical structure between or within documents. Several other dimensionality reduction strategies, most notably **latent semantic indexing** (LSI) (Deerwester et al., 1990), have been proposed by IR researchers to address these drawbacks. LSI performs a singular value decomposition of the X matrix to find a linear subspace in the space of tf-idf features that captures the majority of the variance in the collection. With this method, large collections can experience significant compression. Additionally, Deerwester et al. claim that some fundamental linguistic concepts like synonymy and polysemy can be captured by the derived features of LSI, which are linear combinations of the original tf-idf features.

This model was advanced significantly by Hofmann (1999), who proposed the **probabilistic LSI** (pLSI) model, also referred to as the aspect model, as a replacement for LSI. Each word in a document is modelled by the pLSI approach as a sample from a mixture model, where the mixture components are multinomial random variables that can be thought of as representations of "topics." The pLSI technique models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be seen as representations of "topics." In order to reduce each document to a probability distribution on a predetermined set of topics, each document is represented as a list of mixing proportions for these mixture components. This distribution serves as the document's "reduced description".

The foundation of LSI and pLSI methods is the "bag-of-words" presumption, which holds that the word order within a document can be overlooked. This is referred to as an assumption of exchangeability for the words in a document in the language of probability theory (Aldous, 1985). Furthermore, although less explicitly stated, these methods also presuppose that documents are interchangeable, and the precise ordering of the documents in a corpus can also be overlooked. Any group of exchangeable random variables has an infinite representation as a mixture distribution, according to a well-known representation theorem attributed to de Finetti (1990). As a result, mixture models that account for the exchangeability of both words and documents must be taken into account if we are to consider exchangeable representations for both documents and words. The **Latent Dirichlet allocation (LDA)** model that we present in the current paper results from this line of deductive reasoning.

To be clear, an assumption of exchangeability does not equate to an assumption of independent and uniform distribution of the random variables. Instead, exchangeability is best understood as "conditionally independent and identically distributed," where the conditioning is with respect to a probability distribution's underlying latent parameter.

1.3 Objectives and Scope

(i) This paper has the motive to discuss and explore the Latent Dirichlet Allocation Model and how it is used to perform task of topic modelling

(ii) Collaborative Filtering Systems and Content-based Recommender Systems are the two broad categories into which recommender systems can be divided. A Collaborative Filtering suggests a product based on past preferences of other users with comparable characteristics. A content-based recommender system makes suggestions for products that are comparable to those the user has previously enjoyed. We would concentrate on developing a fundamental content-based recommender system because the provided dataset only contains item data. Using this model and with the help of coherence score, netflix movie reviews will be topic modelled to be further used a content-based recommendation system.

1.4 Methodology Followed

Each of us must have once questioned the source of the recommendations that Netflix, Amazon, and Google provide. We frequently rate products online, and recommender systems use all of the preferences we submit and information we share, whether consciously or unconsciously, to produce actual recommendations.The main goal is to create a topic modelling LDA algorithm-based recommendation system for content.

(i) Understand and explore the LDA model and its distributions which include beta distribution, dirichlet distribution and multinomial distribution.

(ii) The next stage is to clean the data, tokenization, stemming and lemmetization, and further vectorization of the textual data into numerical forms like one-hot vector.

(iii) For a given number of topics, coherence score is calculated. Topic Coherence is a measure of score of a single topic by measuring the degree of semantic similarity between high scoring words in the topic. These metrics help in the differentiation between topics that can be understood semantically and topics that are the result of statistical inference.

(iv) Finally with the best coherence score we will pick the optimal number of topics to build the model. On this model, using the distributions obtained, a content based recommendation system is built.

1.5 Organization of the Thesis

For ease of understanding, this paper has been divided into 6 sections.

Chapter 1 deals with the introduction of the thesis. It contains a brief review of the earlier literature on the problem and how it evolved into LDA. It provides a introduction to the probabilistic model. Last but not least, it includes the thesis's contents as well as the thesis's objectives and scope.

Chapter 2 includes the important probabilistic distributions that is essential to be known before exploring the actual LDA model.

Chapter 3 Contains the representation, terminology and detailed explanation of LDA

Chapter 4 Training the LDA model using gibbs sampling

Chapter 5 Netflix movie corpus

Chapter 6 discusses all facets of the research that was reported in this thesis. Important conclusions and areas for further research are also included.

CHAPTER 2

Distributions and Probabilistic Models

Preface

Mathematics is the foundation of any machine learning algorithm. This chapter focuses on the crucial probabilistic distributions that serve as the foundation for the LDA model, which will be covered in chapter 3.

2.1 Bernoulli and Binomial Distribution

Knowing how data is distributed is essential because it makes it easier dealing with data, regardless of the field—whether it be probability, statistics, data science, machine learning, deep learning, or any other related one.

In the Bernoulli distribution, every experiment is a discrete probability distribution type which poses a question that can only have a yes or no response. To put it another way, the random variable has the option of being 1 with probability p or 0 with probability (1 - p). A Bernoulli trial is the name given to such an experiment. A Bernoulli Distribution can be used to simulate a pass or fail test. **A Bernoulli trial is occurrence of an experiment considering only 2 outcomes.** Let 'p' be a probability of occurrence of an event Z(event=0,1). Therefore the probability mass function would be:

$$P(z) = (p)^z(1-p)^{1-z} \tag{2.1}$$

The outcome of a single Bernoulli trial is represented by the Bernoulli distribution. The number of successes and failures in n independent Bernoulli trials are represented by the **binomial distribution** for any given value of n. The binomial distribution, for instance, represents the number of successes and failures in a lot of n items if a manufactured item is defective with probability p. Sampling from this distribution in particular allows one to count the number of defective items in a sample lot. A different example can be how many heads you get when you flip a coin n times. The binomial distribution's pdf for n independent Bernoulli trials is given by:

$$P(z) = \binom{n}{z}(p)^z(1-p)^{n-z}, \qquad for\ z\ \epsilon\ [0,n] \tag{2.2}$$

The equation represents the probability of an event 'z' occurring with a probability of 'p' out of total of 'N' events. While Binomial analyses the results of numerous trials of a single event, Bernoulli analyses the results of a single trial of the event.

2.2 Beta Distribution

Beta Distribution is known as the probability distribution on 'probabilities'. The probabilities in various scenarios could be modelled using this flexible probability distribution. It helps us model probabilities we do not know about.

We use the Beta distribution when do not know precisely the probability of a certain event. We treat the unknown probability as a random variable and we assign a Beta distribution to it. In simpler words, it defines what would be the probability for an event to have a particular probability. The Beta distribution's domain is restricted to the range between 0 and 1 because it models a probability.

As this distribution models probabilities of probabilities, an example of Bernoulli distribution would lead a better explanation path. Suppose **'W'** is a bernoulli distributed random variable that stand for weather for a particular day. The weather can be "GOOD" with a probability of "0.8" or "BAD" with a probability of "0.2". This suggests that it is a bernoulli distributed random variable.

$$W \sim Bernoulli(\theta)$$

where θ is the probability of weather being good or bad for a particular day. After observing 1 year let θ be found to be equal to 0.7. For the next year, suppose it increased to 0.9. Therefore, the question may arise as to what could be the probability of θ being equal to "0.85" next year? This is where Beta distribution comes into play. As this distribution is modeled on probabilities and its range of input is 0 to 1, it is a continuous distribution.

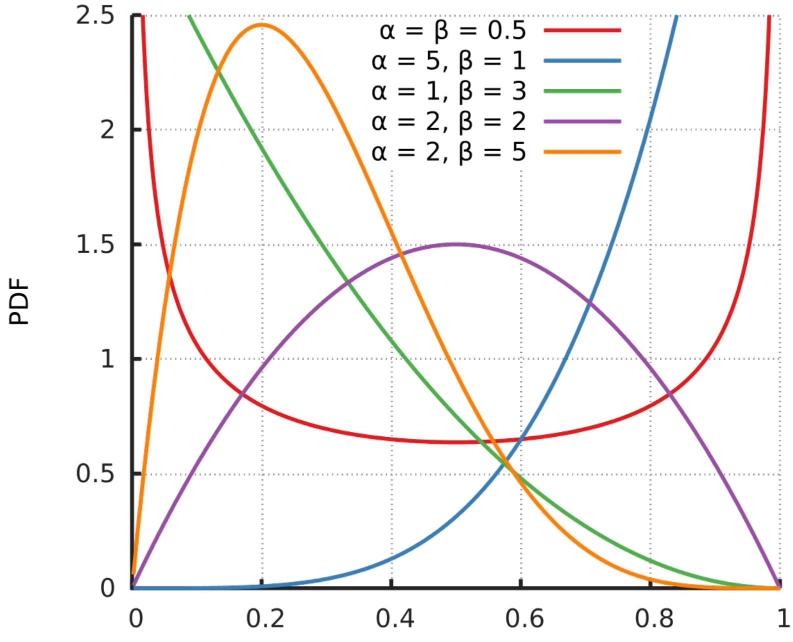


Figure 2.1: Beta Distribution with different values of α and β

As seen in the figure 2.1, the Y axis represents the probability density function and the X axis represents the probability, i.e., value of θ. Here the value can be greater than 1 because it is not normalized yet, we will normalize it with a constant later to bring the probability between 0 and 1. If we consider graph with parameters $\alpha = 2$ and $\beta = 5$ (the orange line), then this graph suggests that there is a higher chance of probability(θ) being around 0.15-0.25.

The probability density function of Beta distribution with parameters (α,β) are:

$$\begin{aligned} \theta &\sim Beta(\theta;\ \alpha, \beta) \\ &\sim (\theta)^{\alpha-1}(1-\theta)^{\beta-1} \end{aligned} \tag{2.3}$$

This suggests that if we chose our parameters to be equal to one, we would get a straight line in beta distribution, which will mean every θ is equally likely. This might not a realistic case.

Furthermore, removing the proportionality and introducing the constant in the equation:

$$Beta(\theta;\ \alpha, \beta) = \frac{(\theta)^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} \tag{2.4}$$

The constant is introduced here to normalize the Y axis values so that the distribution can exhibit probability between 0 and 1. Note that the Beta distribution has a beta as its parameter. The constant introduced in the equation is called **"Beta Function".**

One particular class of function is the beta function, also referred to as the first-class Euler integral. B(x, y) is a common form where x and y are real numbers greater than 0. Additionally, it has a symmetrical form, such as B(x, y) = B(y, x). Special functions is a term used in mathematics for it. The beta function strongly correlates each input value with a single output value. Many mathematical operations heavily rely on the beta function. The beta function is defined as follows:

$$B(p, q) = \int_0^1 t^{p-1}(1-t)^{q-1}\, dt \tag{2.5}$$

Due to its close relationship with the gamma function, which serves as the generalisation of the factorial function, the beta function is crucial to calculus. Many intricate integral functions in calculus can be broken down into simpler normal integrals involving the beta function. Relation with gamma function

$$B(p, q) = \frac{\Gamma p \Gamma q}{\Gamma(p+q)} \tag{2.6}$$

where gamma function defined as:

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \tag{2.7}$$

If p and q are in integers then beta function can also be represented in factorial form as follows:

$$B(p,q) = \frac{(p-1)!(q-1)!}{(p+q-1)!} \tag{2.8}$$

2.3 Categorical Distribution

In probability theory and statistics, a categorical distribution is a discrete probability distribution that specifies the probability of each category separately and describes the potential outcomes of a random variable that can take on one of K possible categories. The generalised form of the Bernoulli distribution is another name for it. Although these results have no inherent ordering, numerical labels are frequently used to describe the distribution (e.g., 1 to K). The most typical distribution over a K-way event is the K-dimensional categorical distribution; any other discrete distribution over a size-K sample space is a special case. Only that each must fall between 0 and 1, and that they all must add up to 1, serves as a constraint on the parameters used to specify the probabilities of each possible outcome. The categorical distribution is the generalisation of the Bernoulli distribution for a discrete variable such as the roll of a die that has more than two possible outcomes. The categorical distribution, on the other hand, is a special case of the multinomial distribution because it provides probabilities of potential outcomes of a single drawing rather than a series of drawings.

In the former section of Beta Distribution it was discussed about a bernoulli distributed random variable 'W' that has two categories, that was "good" or "bad" (like success or failure). But what if the categories are more than 2? Here comes the categorical distribution into play. Now suppose the random variable 'W' is distributed over 3 categories namely: Cloudy(C or 0), Sunny(S or 1) and Rainy(R or 2). Let their probabilites be:

$$P(W=C) = \theta_0, P(W=R) = \theta_1, P(W=R) = \theta_2 \tag{2.9}$$

$$\theta = \{\theta_0, \theta_1, \theta_2\} \tag{2.10}$$

The important constraints over θs is that the sum of all θs must be equal to 1:

$$\sum_{i=0}^{D-1} \theta_i = 1 \tag{2.11}$$

The 'θ' array can be as follows: $[0.2, 0.3, 0.5]$, which means that probability of day being cloudy is '0.2'. The probability mass function can be defined as:

$$P(W) = \prod_{i=0}^{D-1} \theta_i^{I(W=i)} \tag{2.12}$$

'$I(W = i)$' is nothing but indicator function defined as:

$$I(W = i) = \begin{cases} 1, & \text{if } W = i \\ 0, & \text{otherwise} \end{cases} \tag{2.13}$$

The above equation suggests that all probabilities of the weather which are not occurring in the sample space will be equal to '1' except the weather which is occurring.

2.4 Multinomial Distribution

The multinomial distribution in probability theory is an expanding of the binomial distribution. For instance, it simulates probabilities such as counts for every side of a k-sided die rolled n times. The multinomial distribution provides the probability of any specific combination of numbers of successes for the various categories for n independent trials, each of which results in a success for precisely one of k categories, with each category having a given fixed success probability. The multinomial distribution is the Bernoulli distribution when k is 2 and n is 1. The binomial distribution is present when k is 2 and n is greater than 1. When k is greater than 2 and n is 1, the distribution is categorical. To emphasise this four-way relationship (where n determines the prefix and k the suffix), the term "multinoulli" is occasionally used to describe the categorical distribution.

Extending the example taken in categorical distribution of 'W' as a random variable representing 3 states, i.e, cloudy, rainy and sunny with probability distribution as $\theta = \{0.2, 0.3, 0.5\}$. Suppose instead of one day as sample space, we now have a week's observation, $D = \{C, S, S, S, R, C, S\}$, where C=Cloudy, R=Rainy and S=Sunny. Now if we use just the categorical distribution and compute the probability, we would get probability of observation as $(0.2)^2(0.3)^1(0.5)^3$. But the problem is even if we change the order of occurrence of the weather in the given observation, our answer will be same. It means we are missing some permutations or paths that have the same probability distributions. This is Multinomial Distribution.

Suppose we use one-hot vector encoding for the observed dataset, such as, $k = [2, 8, 9]$. This vector represents that the observed data contains 2 cloudy days, 8 rainy days and 9 sunny days. Therefore the Probability Mass Function for multinomial distribution will be:

$$P(k) = \frac{n!}{\prod_{d=0}^{D-1}(k_d)!}\prod_{d=0}^{D-1}\theta_d^{k_d} \tag{2.14}$$

Applications in biology and geology frequently use multinomial distributions. Gregor Mendel, an Austrian botanist, crossed two strains of peas, one with green and wrinkled seeds and the other with yellow and smooth seeds, resulting in strains with four different types of seeds: green and wrinkled, yellow and round, green and round, and yellow and wrinkled. He learned the fundamentals of genetics by studying the resulting multinomial distribution.

2.5 Dirichlet Distribution

In Bayesian statistics and machine learning, the Dirichlet distribution is a continuous probability distribution that is frequently used. It bears the name Johann Peter Gustav Lejeune Dirichlet, a German mathematician. This family of continuous multivariate probability distributions is parameterized by a vector of positive reals. The beta distribution is a univariate distribution, and the Dirichlet distribution is a multivariate generalisation of that distribution. **What the Beta dsitribution is to the Bernoulli/binomial, the Dirichlet distribution is to the categorical/multinomial.** Hence sometimes it is also referred as multi-variate beta distribution (MBD). This is the main distribution used in the LDA model where the topic-document and word-topic are the 2 dirichlet distributions used in the same model.

Lets say D is a 2 state categorical variable with probability distributions $\theta = \{\theta_0, \theta_1\}$. The sum of θ_0 and θ_1 must be equal to 1 as they represent 2 categories probabilities. Therefore the vector or point θ must lie in the line $\theta_0 + \theta_1 = 1$. Refer Fig 2.2.

Now we can plot a probability density function over this line to find out which of these points/probabilities are more likely than the others. This is the Dirichlet distribution. Refer Figure 2.3

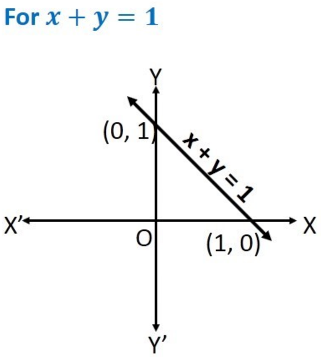


Figure 2.2: All possible values of vector $\theta = \{\theta_0, \theta_1\}$

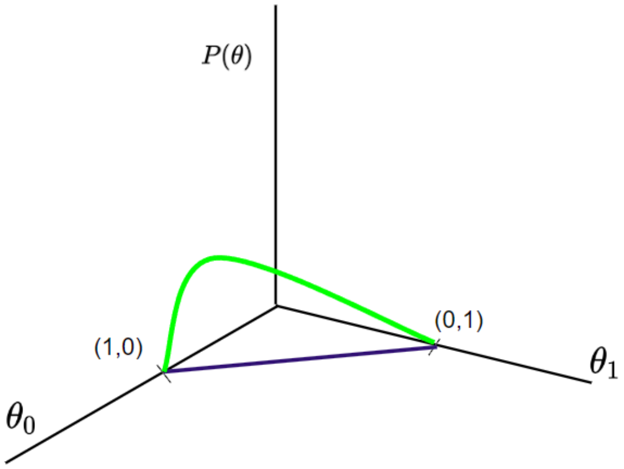


Figure 2.3: Dirichlet distribution over the vector $\theta = \{\theta_0, \theta_1\}$

The probability distribution for the graph 2.3 is proportinal to:

$$P(\theta) \sim {\theta_0}^{\alpha_0 - 1} . {\theta_1}^{\alpha_1 - 1} \qquad ; \alpha_0, \alpha_1 > 0 \tag{2.15}$$

Recall the Beta distribution looked similar to this equation, just the fact that Beta distribution was not defined over a vector θ, it was defined over a scalar θ instead.

$$P(\theta) \sim (\theta)^{\alpha - 1}(1 - \theta)^{\beta - 1} \tag{2.16}$$

In essence, we can state that this was nothing but 2 state categorical Dirichlet distribution.

Now if D=3, i.e., 3 state categorical variable, all the feasible points of the vector θ can be plotted on a 2-simplex as given in figure 2.4

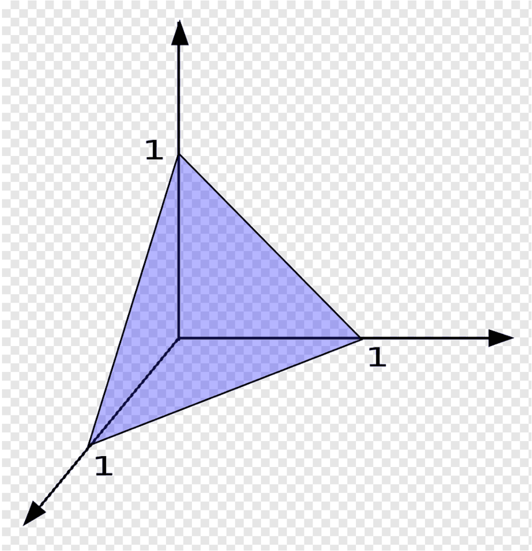


Figure 2.4: 2-simplex

.

The Dirichlet Distribution on the 2 simplex will be proportional to:

$$P(\theta) \sim {\theta_0}^{\alpha_0-1}.{\theta_1}^{\alpha_1-1}.{\theta_2}^{\alpha_2-1} \tag{2.17}$$

If we plot the dirichlet distribution on the 2 simplex with different values of dirichlet parameter α_i, it would look like graphs in figure 2.5.

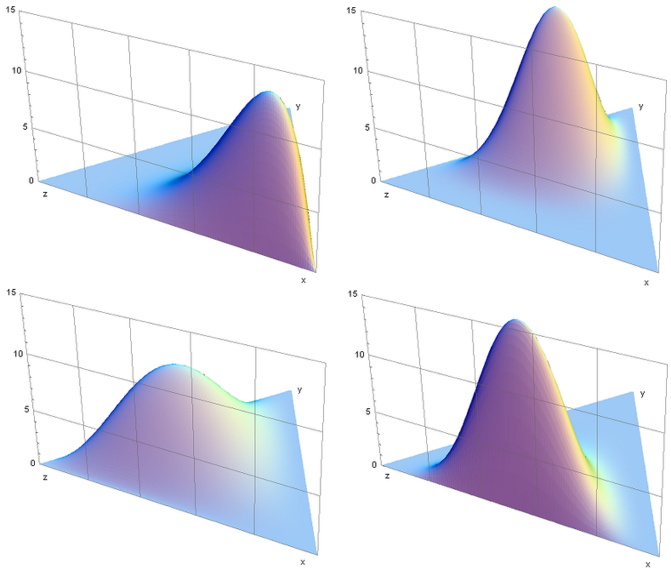


Figure 2.5: Dirichlet Distribution on 3 variable categorical random variable that forms a 2 simplex. The graph shows dirichlet distributions with different values of α

.

In general, let 'W' be a D-state categorical random variable. Therefore let the probabilities θ ϵ D-dimension vector where, $\sum_{d=0}^{D-1}\theta_d = 1$ (sum of all probabilities of categorical variables/states should be equal to 1). A D-state dirichlet must have the values of θ_d (probabilities) lie on a **D-1 dimensional simplex** with corners at $[1, 0, 0, ...]$, $[0, 1, 0, ...]$, $[0, 0, 1, ...]$ and so on. The probability density function is proportional to:

$$P(\theta) \sim \prod_{d=0}^{D-1} \theta_d^{\alpha_d - 1} \tag{2.18}$$

The proportionality is reduced by introducing a constant of Beta function:

$$P(\theta) = \frac{\Gamma(\sum_{d=0}^{D-1} \alpha_d)}{\prod_{d=0}^{D-1} \Gamma(\alpha_d)} \prod_{d=0}^{D-1} \theta_d^{\alpha_d - 1} \tag{2.19}$$

A quite common instance of dirichlet distribution is the symmetric Dirichlet distribution, where each component of the parameter vector 'alpha' has the same value. When a Dirichlet prior over components is required, but there is no prior information favouring one component over another, the symmetric case may be helpful. The symmetric Dirichlet distribution can be parametrized by a single scalar value, referred to as the concentration parameter, because all components of the parameter vector have the same value. The density function has the following form in terms of.

$$\alpha_0 = \alpha_1 = \alpha_2 = ... = \alpha \tag{2.20}$$

The symmetric Dirichlet distribution is uniform over all points in its support when α=1, which is equivalent to a uniform distribution over the open standard (K 1)-simplex. The flat Dirichlet distribution is the name given to this particular distribution. When the concentration parameter is greater than 1, the preferred variates are those with dense, uniform distributions, meaning that all of the values within a sample are similar to one another. Low concentration parameter values favour sparse distributions, where the majority of values within a sample are close to 0 and the majority of the mass is concentrated in a small number of values.

CHAPTER 3

LDA Model

Preface

As all the necessary theories and distributions that forms the base the model has been covered in the previous section, this chapter dives deep into the actual Latent Dirichlet Allocation model and explores how it is used for the task of Topic Modelling.

3.1 Topic Modelling

The use of topic modelling methods in natural language processing for topic discovery and semantic mining from unordered documents is a powerful technique. Topic modelling techniques built on LDA have been widely used in information retrieval, social media analysis, text mining, and natural language processing. For instance, topic modelling based on social media analytics makes it easier to comprehend the responses and discussions among users in online communities, as well as to extract helpful patterns and understandable relationships from their interactions in addition to what they post on social media sites like twitter and Facebook. Topic models are popular for illustrating discrete data and provide a useful method for uncovering hidden semantic structures in vast amounts of data. For the first time, a group of researchers who studied topic modelling in software engineering used LDA to extract topics from source code and visualise software similarity. In other words, LDA is used as a natural method to determine the distribution of each document over topics and calculate the similarity between source files. For the first time, a group of researchers who studied topic modelling in software engineering used LDA to extract topics from source code and visualise software similarity. In other words, LDA is used as a natural method to determine the distribution of each document over topics and calculate the similarity between source files. The authors showed that this method can be useful for software refactoring and project organisation. Topics are nothing but words that appear frequently in statistically significant methods grouped together. A text can be any type of unstructured text, including emails, book chapters, blog posts, journal articles, and blog posts. For topic modelling, Topic models are unable to comprehend the meanings and concepts of words in text documents. Instead, they make the assumption that any portion of the text is put together by choosing words from probable word baskets, where each basket is related to a particular topic. The tool repeats this process until it settles on the most likely distribution of words into topic-specific baskets. In terms of the collection as a whole, the individual documents, and relationships between the documents, Topic modelling can offer a useful view of a large collection.

3.2 LDA Model

LDA is a probabilistic generative model of a corpus. A topic is defined by a distribution over words, and the documents are conceptualised as random mixtures over latent topics. One of the most widely used techniques in topic modelling is latent Dirichlet allocation (LDA), which was first presented by Blei, Ng, and Jordan in 2003. LDA uses word probabilities to represent topics.

The words in each topic with the highest probabilities typically provide a good indication of the topic's nature. The underlying premise of LDA is that each document can be represented as a probabilistic distribution over latent topics, and that the topic distributions across all documents have a common Dirichlet prior. The word distributions of the latent topics in the LDA model also share a common Dirichlet prior, and each latent topic is represented as a probabilistic distribution over words. We can infer that LDA is a generative model from the figure 3.1.

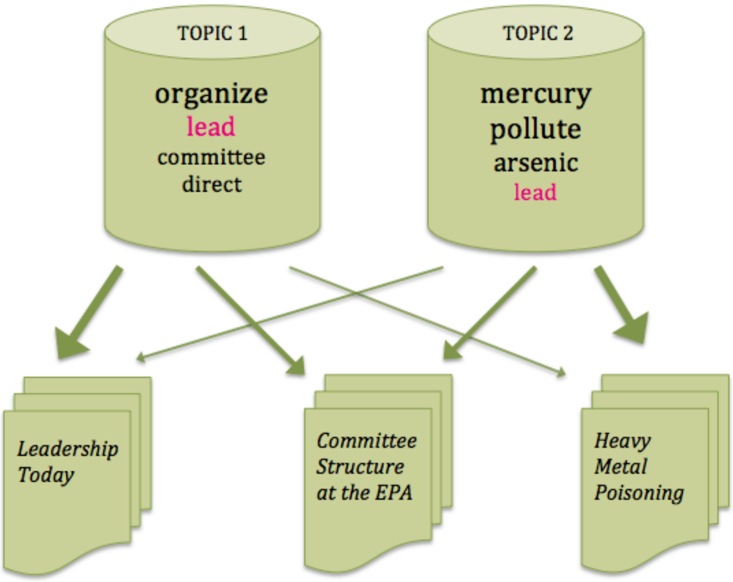


Figure 3.1: Example of how LDA generates documents from clustered topics according to their probability distribution

.

The figure 3.1 shows that LDA initially needs to build the topics by topic-word distribution. From this distribution, the model can generate documents by finalizing what percent or probabilities of different topics must be present in each document, thus generating the entire document corpus. Some basic assumptions of topic modelling using LDA are:

- The word groups used in documents with related topics are similar to each other.
- Then, by looking for word groups that recur frequently together in documents throughout the corpus, latent topics can be discovered.
- Documents are probability distributions over latent topics, meaning that a given document is likely to contain more words that are related to a particular topic.
- Topics are a probability distribution over words within themselves.

3.3 Graphical Notation of LDA

LDA model is graphically represented in plate notation for better understanding of how all the components of the model are interacting with each other. The graphical notation can be interpreted as figure 3.2.

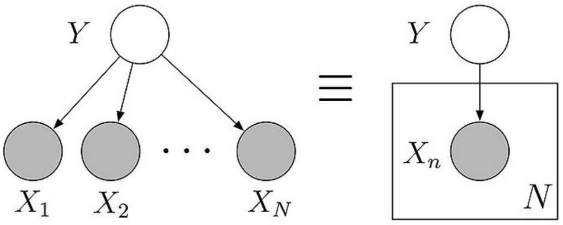


Figure 3.2: Interpretation of graphical notation

.

Some points to observe in the above graphical notation are:

- X_1, X_2, ..., X_N, Y are all random variables.
- Edges in the graph denote possible dependence between variables (Conditional Dependence).
- Observed variables like X_1, X_2, ..., X_N are shaded.

The Graphical Notation of LDA model is(figure 3.3): .

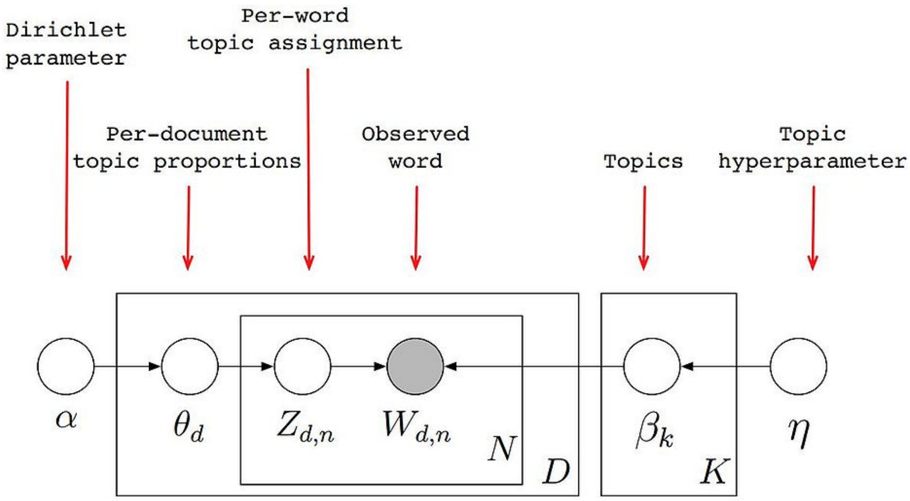


Figure 3.3: Graphical Notation of LDA Model

Everything in the figure is hidden or latent except the shaded variables. Formally the model is defined as follows:

- There are 'K' different topics and each β_i, $(i = 1 : K)$, is the probability distribution over all the words for each topic 1 to K. Each β_i has the probabilities of words that can happen or appear in topic 'i'. **Therefore it is the distribution over words for a given topic.** (It is a real vector of length 'V' where 'V' is the lenght of vocabulary for our corpus)

- Total number of documents in the corpus is given to be 'D'. For every document 'd', we have θ_d variables that represent per-document proportions or probabilities of topics. In other words, θ_d represents at what probabilities topics are present in document 'd'. θ_d **is the distribution over topics for a given document**

- A document is visualised as a collections of 'N' words denoted by $w = \{w_1,\ w_2,\ ...,\ w_N\}$ where w_n is the nth word in sequence in the document.

- $Z_{d,n}$ is per word topic assignment, each word will be assigned only 1 topic. It is the topic for nth word in dth document. Before generating the actual document, first the model will assign a sequence of topics to each word space (where the actual word will be generated later). Same words can be assigned different topics at different locations, but for a particular instance only 1 topic will be assigned.

- Finally $W_{d,n}$ is the actual word generated from the topic sampled from $Z_{d,n}$. If we have the accurate hyper parameters for our distribution, we will be able to generate all the documents in the corpus. That is our final goal.

The discrete random variables are distributed by following probability distributions:

$$p(Z_{d,n} = k|\theta_d) = (\theta_d)_k \tag{3.1}$$

$$p(W_{d,n} = v|Z_{d,n}, \beta_1, \beta_2, ..., \beta_K) = (\beta_{Z_{d,n}})_v \tag{3.2}$$

Here $(\theta_d)_k$ is the kth element of the vector θ_d which corresponds to the percentage of document d corresponding to topic k. Equation 3.1 is the probability of nth word in dth document($Z_{d,n}$) being assigned to a topic 'k' given the distribution of topics θ_d over the document d. Similary equation 3.2 is the probability of a word 'v' from vocabulary of word of size

'V' being assigned to nth word in dth($W_{d,n}$) document given that word should be from a topic $Z_{d,n}$ along with all the topic distributions.

Given the parameters α and η, the joint distribution mixture of a topic mixture θ, a set of N topics Z, and a set of N words W is given by:

$$p(\theta, Z, W|\alpha, \eta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \eta) \tag{3.3}$$

where $p(z_n)$ is simply θ_i for the unique i such that z_i^n. Integrating over θ and summing over z, we obtain the marginal distribution of a document:

$$p(W|\alpha, \eta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \eta) \right) d\theta \tag{3.4}$$

The joint distribution of a single document can be also viewed as follows where α and η are dirichlet parameters.

$$p(w, z, \theta, \beta|\alpha, \eta) = \prod_{k=1}^{K} p(\beta_k|\eta) \prod_{d=1}^{D} \left(p(\theta_d|\alpha) \prod_{n=1}^{N} p(z_{d,n}|\theta_d) p(w_{d,n}|z_{d,n}, \beta) \right) \tag{3.5}$$

Finally, by adding the marginal probabilities of individual documents, we can calculate the probability of a corpus(M is the number of documents and N is size of document):

$$p(D|\alpha, \eta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{d,n}} p(z_{d,n}|\theta_d) p(w_{d,n}|z_{d,n}, \eta) \right) d\theta_d \tag{3.6}$$

3.4 Generative Model

As the above notation suggests that our aim is to find the right parameters to tune our distributions so that it will generate documents that will be closely identical to the original documents we initially had. If we have the right parameters, we can then use the topic distributions and cluster the given documents according to their latent properties. The topic distribution θ and the word distribution β both are Dirichlet Distribution as discussed in the section 2.5. The topics that are sampled for each document, for each word space are sampled from a multinomial distribution over topics and words. In LDA, words are observed, topic and word distributions are hidden, and α and η are the hyperparameters. Thus, we need to infer the distributions and the hyperparameters. The algorithm of LDA in a nutshell is:

- We sample each topic distribution $\beta_i \ \sim Dir(\eta)$, for every topic i ϵ {1, ..., K}.
- For each document 'd':
  - Draw topic proportions $\theta_d \ \sim Dir(\alpha)$
  - Choose document length $N_d \sim Poisson(\xi)$
  - For each word space:
    - Draw $Z_{d,n} \ \sim Mult(\theta_d)$
    - Draw $W_{d,n} \ \sim Mult(\beta_{Z_{d,n}})$

For each document, draw the topic proportions, i.e., what all topics are involved in these document as per the Dirichlet distribution with parameter α. As we are going to generate a document consisting of words that would not necessarily mean anything if we read them, they would be just a collection of words. These collection of words are going to be matched with the original text pre-processed documents to find best closely approximation of the given document. For instance:

$$\begin{aligned} &T_1- > \beta_1 = \{0.010, 0.001, ..., 0.050\} \\ &T_2- > \beta_2 = \{0.000, 0.030, ..., 0.001\} \\ &... \\ &T_K- > \beta_K = \{0.230, 0.004, ..., 0.032\} \end{aligned} \tag{3.7}$$

The vectors represent probability distribution of words $\{w_1, w_2, ...\}$ for the particular topic. The Dirichlet parameter 'η' defines what kind of distribution will be preferred over the others for a given topic. A similar Dirichlet distribution, that is θ with parameter α, is performed over topics for a given document. These θ_i vectors represent how the topics are distributed in every single document 'i'. The α parameter suggests what kind of distribution of topics will be preferred for a document than the others. We have a collection of documents which came from a particular dataset or a random source. In order to learn the topic representation of each document and the words associated with each topic, we will choose a fixed number of K topics to discover and apply LDA to do so. Each document is cycled through by the LDA algorithm, and each word is randomly assigned to one of K topics. This random assignment already provides word distribution for all topics, topic representation for all documents, and word distribution for all documents. To make these topics better, LDA will backtrack each and every word in every document. We eventually reach a roughly steady state where the assignments are acceptable after repeatedly iterating the backtracking step. Each document is given a topic at the conclusion. We can look for the words that are most likely to be associated with a particular topic.

The dirichlet distribution which are used to draw the per-document topic distribution and per-topic word distribution can be altered by changing the value of parameter α and η respectively. Suppose, $\theta \sim Dir(\alpha)$ is a dirichlet distribution on a parameter α on a 2-simplex(figure 2.4). Let there be 2 probability distribution vectors $\theta_1 = \{0.98, 0.01, 0.01\}$ and $\theta_2 = \{0.33, 0.33, 0.34\}$. If we consider this 2 simplex representing 3 topics(for every 3 corner of triangle), then by looking at the θ distribution we can claim that θ_1 distribution is dominant on 1 topic while the θ_2 is equally distributed among the 3 topics.

For first case let the value of $\alpha = 0.1$, i.e., $\alpha < 1$. Therefore according to the probability density function of dirichlet distribution the value of probabilities of θ_1 and θ_2 are:

$$\begin{aligned}
&p(\theta|\alpha_0, \alpha_1, \alpha_2) \sim {\theta_0}^{\alpha_0 - 1}.{\theta_1}^{\alpha_1 - 1}.{\theta_2}^{\alpha_2 - 1} \\
&p(\theta_1|\alpha) \sim (0.98)^{0.1-1}(0.01)^{0.1-1}(0.01)^{0.1-1} = \mathbf{4052.362} \\
&p(\theta_2|\alpha) \sim (0.33)^{0.1-1}(0.33)^{0.1-1}(0.34)^{0.1-1} = \mathbf{19.41}
\end{aligned} \tag{3.8}$$

By putting the values in the equation we can clearly see that if the value of α is less than 1, **the Dirichlet Distribution will have a more likelihood for points in simplex that are dominant in one topic and not in others** rather than those points which are equally distributed in all the topics. θ_1 clearly has higher probability in the dirichlet distribution than θ_2.

For the next case let the value of $\alpha = 2$, i.e., $\alpha > 1$, then the probability distribution will look like:

$$\begin{aligned}
&p(\theta_1|\alpha) \sim (0.98)^{2-1}(0.01)^{2-1}(0.01)^{2-1} = \mathbf{0.000098} \\
&p(\theta_2|\alpha) \sim (0.33)^{2-1}(0.33)^{2-1}(0.34)^{2-1} = \mathbf{0.037026}
\end{aligned} \tag{3.9}$$

As we can see now the probability of θ_2 is higher than θ_1 and therefore as expected, when the value of $\alpha > 1$, then the dirichlet distribution tends to higher probability towards the points which are equally distributed in all the 3 topics of 2-simplex.

By looking at this instance, we can infer that the parameter of the dirichlet distribution can be altered to make the distribution tend towards points or vector of probabilities(θs) which are more dominant in one topic and less dominant in others. If $\alpha > 1$, $Dir(\alpha)$ will prefer distribution where all the topics are equally distributed. On the other hand if $\alpha < 1$, there is a bias to pick topic distributions favouring just few topics. This can be visualised in the figure 3.4:

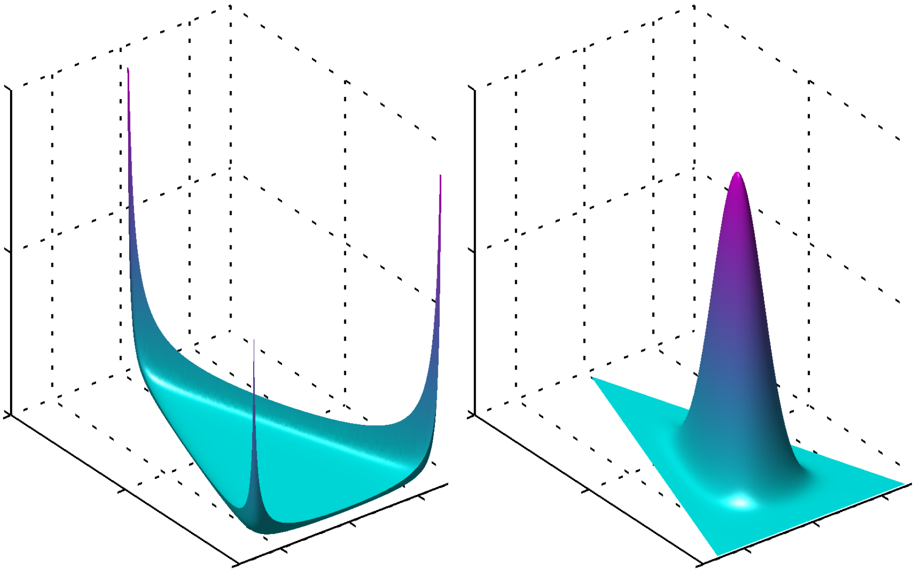


Figure 3.4: Left dirichlet distribution has a parameter value < 1 and Right dirichlet distribution has a parameter value > 1

.

CHAPTER 4

Gibbs Sampling

Preface

In this section we are going to discuss the inference of the LDA model, what parameters to choose so that LDA generates the original documents and how to train the LDA model.

4.1 Inference

We discussed about LDA being a generative model, but now it is time to switch the problem in opposite direction. What if we have a corpus of documents and we want to extract topics? To solve this issue we will be working under the assumption that the documents were generated using a generative model like the one we discussed in former sections. Under this assumption we need to finalize the parameters of the model to generate the original set of documents with close approximation. Numerous techniques, including the variational method, expectation propagation, and Gibbs sampling, have been proposed to estimate LDA parameters.

- Expectation-Maximization (EM), an effective method for estimating graphical model parameters, can be used for unsupervised learning. In fact, the algorithm relies on determining the parameter estimates with the greatest likelihood when the data model depends on particular latent variables. The E-step (expectation) and the M-step (maximisation) are the two steps in the EM algorithm.

- Variational Bayes inference (VB), which utilises a parametric approximation to the posterior distribution of both parameters and other latent variables, can be thought of as a type of EM extension that aims to maximise the fit (for example, using KL-divergence) to the observed data.

4.2 Gibbs Sampling

Finding the topics that are prevalent in a document is an intriguing problem in natural language processing (NLP), which can be solved using Gibbs sampling. It is assumed that there is a fixed vocabulary (made up of V distinct terms) and K distinct topics, each of which is represented as a probability distribution (β_k over the vocabulary) with a distinct Dirichlet prior η. We need to build:

- Topic distributions β_i for i={1, ..., K}

- Per document topic proportions 'θ_d' for document 'd'.

- Per document per word topic assignment $Z_{d,n}$, based on the given observed corpus.

This is a Markov chain Monte Carlo method which simulates a high dimensional distribution by sampling on lower dimensional subset of variables where each subset is conditioned on the value of all others. A process for creating a sample from a joint distribution when it is only practical to compute the conditional distributions of each variable. Sampling is done sequentially and proceeds until sampled values approximate the target distribution. We do not compute joint probability of everything and try to maximize it at once, rather we assume some to be known and by conditioning on this we find the probability of others. It directly estimates the posterior distribution over Z and uses this to provide estimates for β and θ.

Suppose we have a word token 'i' for which we want to find topic assignment probability. $p(Z_i = j)$ is the probability that token 'i' belongs to a topic 'j'. We represent the collection of documents by a set of word indices w_i and document indices d_i for this token i. Gibbs Sampling considers each word token in turn and estimates the probability of assigning the current word token to each topic, conditioned on the topic assignment to all other word tokens(each iteration).

$$p(Z_i = j|Z_{-i}, w_i, d_i) \tag{4.1}$$

The above equation tells the probability of token 'i' having topic 'j' given all other tokens, apart from i, is correctly assigned topic assumed for a given word and given document. The probability of a word being given a topic 'd' should depend on 2 things.

- How likely is topic 'j' to be assigned to 'd_i'?
- How likely is word 'w_i' is for topic 'j'?

$p(Z_i = j)$: the probability of topic 'j' being assigned to a word token Z_i is done for all the topics, i.e., j $\epsilon\{1, ..., K\}$. Therefore it will give us a multinomial distribution for every token Z_i. Therefore this will be done for every token, that is we will sample one topic from this multinomial distribution and assign it to each token. We will repeat these steps and store samples as Gibbs samples.

For this we need to compute 2 different matrices C^{WT} and C^{DT}. C^{WT} has the dimensions as words(W) X Topics(T) and C^{DT} has the dimensions as Documents(D) X Topics(T). C^{WT}_{wj} contains the number of times word w is assigned to topic j, not including the current instance. C^{DT}_{dj} contains the number of times topic j is assigned to some word token in document d, not including the current instance. The probability of the token can be calculated as:

$$p(Z_i = j|Z_{-i}, w_i, d_i) \sim \left(\frac{C^{WT}_{w_ij} + \eta}{\sum_w C^{WT}_{wj} + W\eta}\right)\left(\frac{C^{DT}_{d_ij} + \alpha}{\sum_t C^{DT}_{dt} + T\alpha}\right) \tag{4.2}$$

The left part is the probability of word 'w' under topic 'j'.(how likely a word is for topic 'j'). The right part is the probability of a topic 'j' under the current topic distribution for document 'd' (how dominant a topic is in a document). The denominator $\sum_w C^{WT}_{wj}$ is the summation of number of words in the topic 'j' and $\sum_t C^{DT}_{dt}$ is the summation of all topics in document 'd'. Now we have the multinomial distribution for every word token and we will sample a topic from here. The η and α are nothing but the actual dirichlet parameters that we will be using to run the LDA model. These act as the smoothing parameters. If the value of these parameters is high, the numerator sum will not matter and all the topics will be assigned roughly equal probabilities. And if the parameter value is low, some topics might dominate the other over a token. Algorithm in a nutshell:

- Each word token is assigned to a random topic from 1 to K.
- Then we compute our matrices C^{WT} and C^{DT}.
- Now for each word token, a new topic is sampled as per $p(Z_i = j|Z_{-i}, w_i, d_i)$, parallely adjusting the matrices C^{WT} and C^{DT} at every iteration.
- A single pass through all the word tokens is called one Gibbs Sample. The initial samples are not saved as they may be inaccurate. The initial samples are called the burnin period. After this burnin period, these samples are saved at regular spaced intervals to prevent correlation between samples.

We can formulate our β and θ distributions by taking expectation over these regular spaced interval samples.

$$\beta_i(j) = \frac{C^{WT}_{ij} + \eta}{\sum_{k=1}^{W} C^{WT}_{kj} + W\eta} \tag{4.3}$$

The $\beta_i(j)$ is the probability that topic 'j' is assigned to the ith word.

$$\theta_j(d) = \frac{C^{DT}_{dj} + \alpha}{\sum_{k=1}^{K} C^{DT}_{dk} + T\alpha} \tag{4.4}$$

The $\theta_j(d)$ is the probability of topic 'j' being in the document 'd'.

4.3 Evaluation

The K topics are represented as multinomial distributions over V after roughly approximating the posterior distribution of the LDA. Each topic distribution includes every word but gives each word a different probability. High probability topics contain words that are more likely to occur together. The top 10 or top 15 high-probability words are typically used to interpret and semantically label the topics. However, LDA generates as many topics as K defines; a low K yields too few or very broad topics, whereas a high K yields confusing topics or topics that should have been combined in the ideal case. Thus, selecting the appropriate value of K is a crucial step in topic modelling algorithms besides LDA.

4.3.1 Topic Coherence

Coherence measures have been proposed by some researchers in the Natural Language Processing community to assess topics. Even though the topics that topic models teach students frequently seem useful, this isn't always the case. Automatically measuring topic coherence makes it easier to quickly spot "junk" topics that may have strong statistical support but have no practical value to end users. This may result in improved methods of interacting with and examining the data, such as information retrieval applications. By measuring the degree of semantic similarity between high-scoring words in a given topic for a given set of documents, a **topic coherence** measure assigns a score to each topic. These metrics aid in separating topics that can be understood semantically from topics that are the result of statistical inference. The mean of a specific coherence score for each pair of words is used to define topic coherence:

$$Topic\ Coherence(z, D) = mean\{score(w_i, w_j, \epsilon)\} \tag{4.5}$$

where z is a topic(collection of words in topic z), D is a document collection and score is a measure of coherence between pair of words in topic z. 'w_i' and 'w_j' represents a pair of words that describes the topic($w_i \ \epsilon \ V$, $w_j \ \epsilon \ V$, $i, j \ \epsilon \ \{1, ..., 10\}$ and $i \neq j$). Depending on the characteristics of the dataset, the term epsilon can be used as a smoothing value to avoid extreme values. As a result of this smoothing, real values are also guaranteed.

Two coherence measures created for LDA that have both been shown to correlate well with human assessments of topic quality are taken into account for our evaluations: The UMass measure (Mimno et al., 2011) and the UCI measure (Newman et al., 2010) are two examples.

UCI Measure

Pointwise Mutual Information (PMI) has been implemented in computational linguistics to determine word associations and word sense disambiguation. PMI calculates the degree to which one variable predicts another.

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{4.6}$$

where the probability of seeing words w_i and w_j together is compared to the probability of seeing them independently. Newman et al. (2010b) defined the UCI measure as follows, deriving it from PMI:

$$Score_{UCI}(w_i, w_j) = \log \frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)} \tag{4.7}$$

We use either the dataset used to train the model, known as Intrinsic, or an external reference dataset, known as Extrinsic, to estimate probabilities. Extrinsic coherence informs us of the coherence of the topics the model has learned based on outside references. When choosing an external dataset, Wikipedia is usually a wise choice because of the wide range of topics it covers. The external dataset can be anything related to the data domain used to build the topic model. Researchers have found that using Wikipedia as an external reference produces the best results. Instead of using an external dataset to compute probabilities, intrinsic evaluation uses the original dataset. It aims to verify that the topics and words the model chose are actually present in the data set. For instance, probabilities would be determined as follows using Wikipedia as a reference:

$$p(w_i) = \frac{D_{wikipedia}(w_i)}{D_{wikipedia}} \tag{4.8}$$

and

$$p(w_i, w_j) = \frac{D_{wikipedia}(w_i, w_j)}{D_{wikipedia}} \tag{4.9}$$

where the number of documents in the entire collection of Wikipedia entries is counted by $D_{Wikipedia}$. $D_{Wikipedia}(w_i, w_j)$ counts the occurrence of the words w_i and w_j at the same entry while$D_{Wikipedia}(wi)$ counts entries at Wikipedia that contain only the word w_i. UCI can be used as an external resource to compare the words in a given document with a preexisting set of topics and words that have gathered a body of subjective semantic evaluations.

UMass Measure

UMass uses conditional probability to calculate the correlation of words in a given document. This idea was used by Mimno et al. (2011) to suggest the UMass measure. This is how its equation is defined:

$$Score_{UMass}(w_i, w_j) = \log \frac{D(w_i, w_j) + \epsilon}{D(w_i)} \tag{4.10}$$

where $D(w_i, w_j)$ counts the number of documents that contain the words w_i and w_j and $D(w_i)$ counts the number of documents containing w_i. The order of the arguments matters according to the asymmetric pairwise score used by UMass. It follows that w_i must be more prevalent than w_j.

The coherence score cannot be judged in just one manner as good or bad. The data that are used to calculate the score determine its value. For instance, a score of 0.5 might be acceptable in one situation but not in another. The only rule is that we should try to get the highest possible score. The coherence rating will typically rise as the number of topics increases. As the number of topics rises, this increase will get smaller. The so-called elbow technique can be used to strike a balance between the number of topics and coherence score. According to the method, coherence score should be plotted as a function of the number of topics. The number of topics is chosen using the elbow of the curve. The goal of this approach is to identify a threshold beyond which further growth in the number of topics is not justified by the diminishing increase in coherence score. Intrinsic and extrinsic measures complement each other in context to topic coherence analysis. Intrinsic measures tell to what extent the words representing a particular topic have in common without any source beyond the original training dataset. In contrast extrinsic measures, quantifies if there is any semantic meaning between the words that represent a topic using external data.

CHAPTER 5

Content Based Recommendation System on Netflix Dataset

Preface

In this chapter we are going to explore the topic modelling on a real world dataset of netflix movies and TV shows. By applying the LDA model on the dataset, extracting the topics and clustering the movies and tv shows with similar genre, we are going to create a content based Recommendation system for users.

Each of us must have once questioned the source of the recommendations that Netflix, Amazon, and Google provide. We frequently review things online, and recommendation systems employ all of the preferences we declare and information we provide, whether consciously or unconsciously, to produce actual recommendations. In order to better grasp the distinctions between the two primary categories of recommendation systems—collaborative or content-based filters—let's take a closer look at a few examples. **Collaborative Filters** are based on user ratings; this form of filter will suggest films to us that we haven't seen but that users who are similar to us have and like. This filter takes into account the movies that both users have seen and how they rated them in order to determine if two users are similar or not. This kind of algorithm will essentially forecast the rating of a movie for a person who hasn't watched it yet based on the similar users' rates by looking at the objects in common. **Content Based** systems do not depend on data obtained by other users. The system will simply choose goods with similar content to recommend to us based on what we like. Less variety will be offered in the recommendations in this instance, but the system will still function whether the user rates items or not.

5.1 Dataset Description

Reed Hastings and Marc Randolph created Netflix, Inc. in Scotts Valley, California, in 1997. The company is an American media services and production corporation with its headquarters in Los Gatos, California. The company's subscription-based streaming service, which provides online streaming of a library of films and television shows, including those created in-house, is its main source of revenue. Over 182 million paid Netflix subscriptions were active as of April 2020, including 69 million in the US. Members have unlimited access to watch on any screen that is connected to the internet at any time. Without interruptions or obligations, members can play, pause, and resume watching at any time.

The dataset consists of 12 columns with 6237 rows of movies and TV shows containing show-id, title, director, country, description, etc.

5.2 Text Preprocessing and Cleaning

Some basic prepocessing steps when dealing with textual data is applied on the dataset to later feed into the LDA model:

- **Stopwords Removal:** Stop words, which are the most common words in any language with no meaning, are often disregarded by NLP. English stop words include "a," "and," "the," and "of." Stop words are frequently eliminated from texts in NLP before they are processed for analysis. In order to streamline the content and omit extraneous details, this is done. Stop words have significant advantages for NLP algorithms. A stop word list, for instance, can be used to accurately filter out stop words when identifying the most frequent word in a sentence.

- **Bi-grams:** A sequence of two contiguous components from a string of tokens—typically letters, syllables, or words—is known as a bigram or digram. When n=2, a n-gram is a bigram. The majority of effective language models for speech recognition use bigrams in addition to other n-grams. Some English words are more frequently used in combination. Sky High, Do or Die, the best performance, a severe downpour, etc. are some examples. Therefore, it may be necessary to find a pair of words in a text document that will aid in sentiment analysis. In order to create these word pairings, we must first retain the present sentence's sentence structure. Bigrams are these kinds of pairs. To assist us generate these pairs, Python's NLTK library includes a bigram function.

- **Lemmetization:** The resulting lemma or token generated through the process conveys the same meaning as the original word and is a recognised word. The lemma of the words "running" and "cats," for instance, is "run" and "cat," respectively. Natural language processing (NLP) frequently uses lemmatization to normalise text data, making it simpler to handle and analyse. Lemmatization takes into account the context and part of speech (POS) of the word, producing a more accurate and relevant lemma. It is comparable to stemming, which likewise seeks to reduce words to their root form.

5.3 Results and Observation

Now that we have brought in the description column, we can begin to deploy our LDA model. Here, we will just be dealing with the description column, and our goal is to determine whether or not we can successfully extract topics from it. Initially we built the LDA model with number of topics randomly chosen to 10 and computed the coherence score of the model. The coherence score came out to be **0.3405...**. As discussed about the coherence score in the former sections, there is no good or bad coherence score as it all depends on the diversity of text and topics in the given dataset. The only objective we must have is to maximize this score by keeping an eye on the trade-off between the coherence score and number of topics. Then coherence scores for number of topics from 5 to 40 in regular intervals of 5, was calculated and plotted against the particular number of topics in figure 5.1:

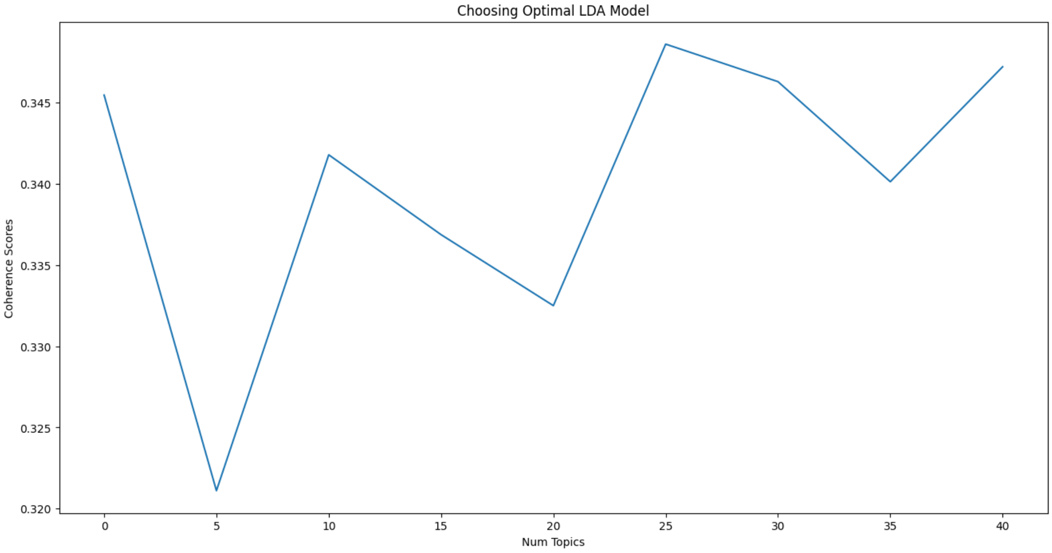
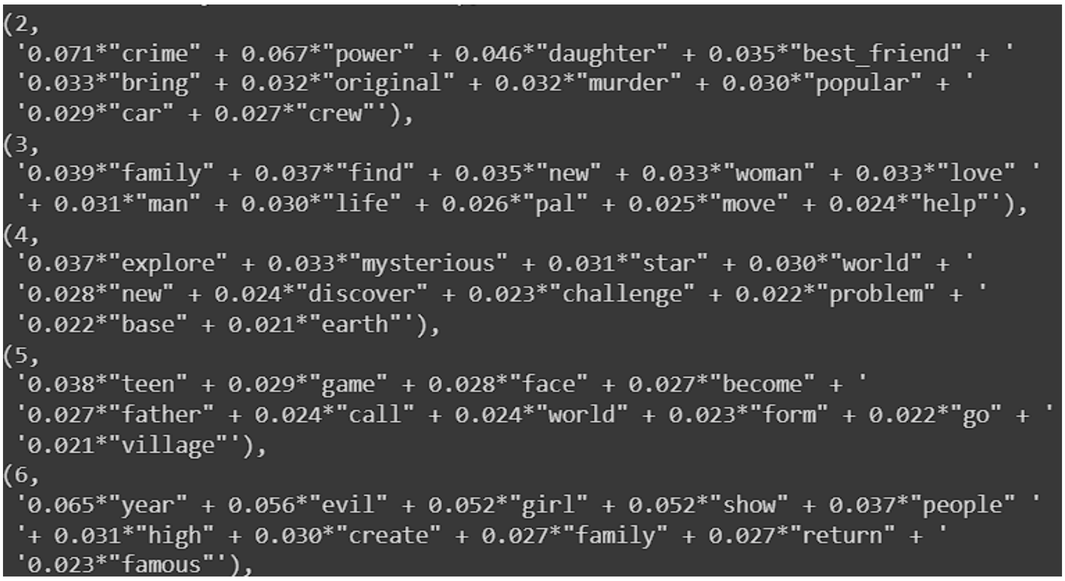


Figure 5.1: Coherence Score plotted against the number of topics for the LDA model

From the above graph we can infer that the coherence scores are highest when number of topic is 1 but that will not be feasible. Therefore as per the elbow method, we can see that there is a peak around where the number of topic is equal to 15. After the building the LDA model around 15 topics, the distribution looked like this(figure 5.2):



```
(2,
 '0.071*"crime" + 0.067*"power" + 0.046*"daughter" + 0.035*"best_friend" + '
 '0.033*"bring" + 0.032*"original" + 0.032*"murder" + 0.030*"popular" + '
 '0.029*"car" + 0.027*"crew"'),
(3,
 '0.039*"family" + 0.037*"find" + 0.035*"new" + 0.033*"woman" + 0.033*"love" '
 '+ 0.031*"man" + 0.030*"life" + 0.026*"pal" + 0.025*"move" + 0.024*"help"'),
(4,
 '0.037*"explore" + 0.033*"mysterious" + 0.031*"star" + 0.030*"world" + '
 '0.028*"new" + 0.024*"discover" + 0.023*"challenge" + 0.022*"problem" + '
 '0.022*"base" + 0.021*"earth"'),
(5,
 '0.038*"teen" + 0.029*"game" + 0.028*"face" + 0.027*"become" + '
 '0.027*"father" + 0.024*"call" + 0.024*"world" + 0.023*"form" + 0.022*"go" + '
 '0.021*"village"'),
(6,
 '0.065*"year" + 0.056*"evil" + 0.052*"girl" + 0.052*"show" + 0.037*"people" '
 '+ 0.031*"high" + 0.030*"create" + 0.027*"family" + 0.027*"return" + '
 '0.023*"famous"'),
```

Figure 5.2: Topics and the probabilities of words being in the topic after building the model

Word clouds, also known as tag clouds, are visual representations of word frequency that give words that appear more frequently in a source text more emphasis. The term's frequency in the document(s) was indicated by the size of the word in the image. By highlighting words that commonly appear in a collection of interviews, documents, or other material, this kind of visualisation can help assessors with exploratory textual analysis. Additionally, it can be utilised to convey the most important ideas or themes throughout the reporting phase. Word clouds for the dataset can be seen in figure 5.3.

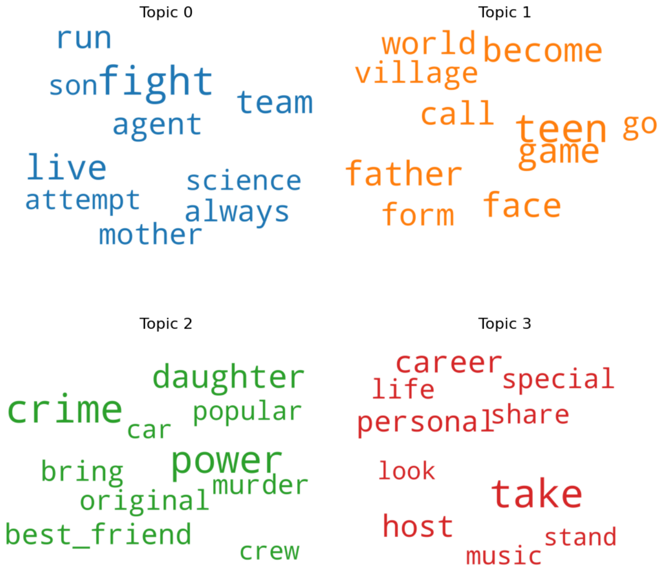
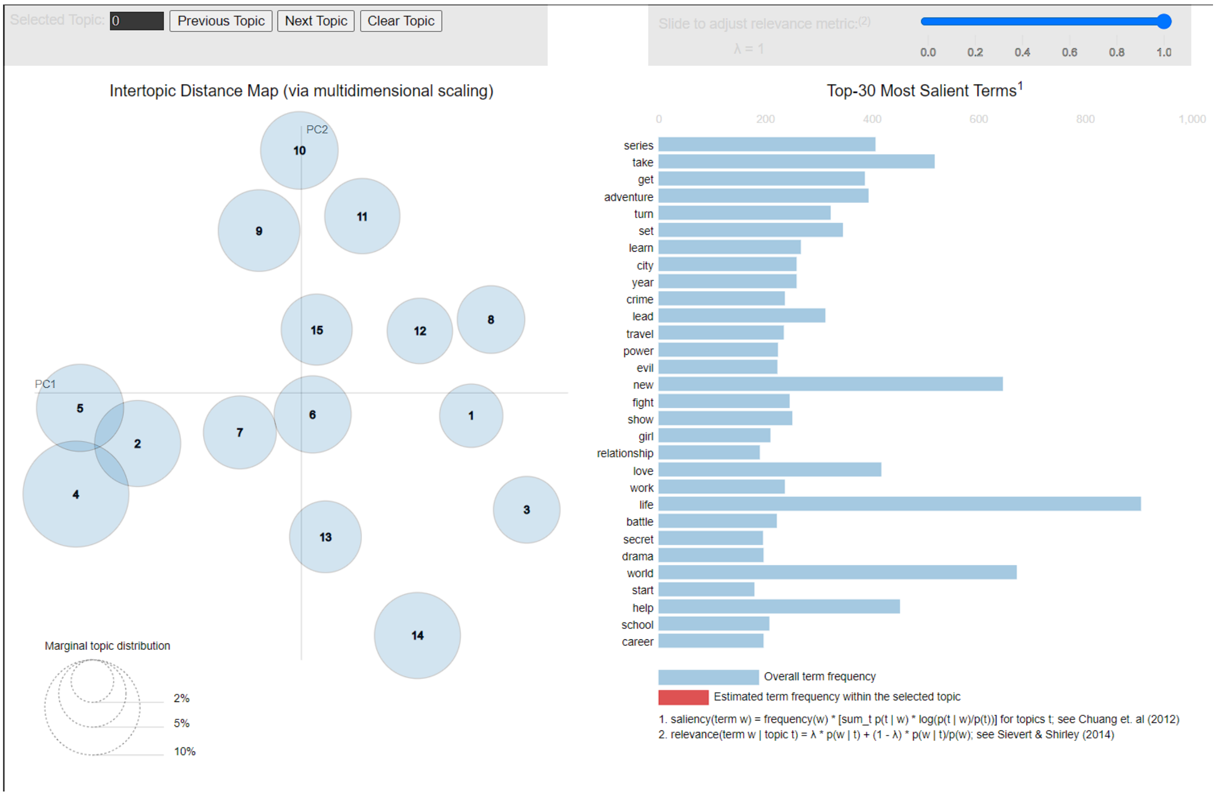


Figure 5.3: Word Cloud of few topics

From the word cloud we may assume vaguely that topic 0 is intending towards something science-fiction genre whereas topic 2 suggests something of a criminal documentary. As the topic modelling task is unsupervised, this makes it quite subjective to specific users to interpret.

The topics are represented in a two-dimensional space by the **intertopic distance map** (figure 5.4). The size of these topic circles reflects the number of words in the English language that are associated with each issue. The circles are displayed using a multidimensional scaling technique based on the words they include, so topics that are closer together have more words in common. This process reduces a large number of dimensions, more than we can comprehend with our human minds, to a sensible number of dimensions, like two. The 30 most important phrases are displayed in the bar chart by default. The bars show the term's overall frequency over the entire corpus. A unique metric called salient, which is explained at the bottom of the visualisation, is used to find the texts in the entire corpus that contain the most valuable or informative words for identifying topics. A word's saliency value tells us how useful it is for recognising a certain topic.



Selected Topic: 0

Slide to adjust relevance metric:(2) λ = 1

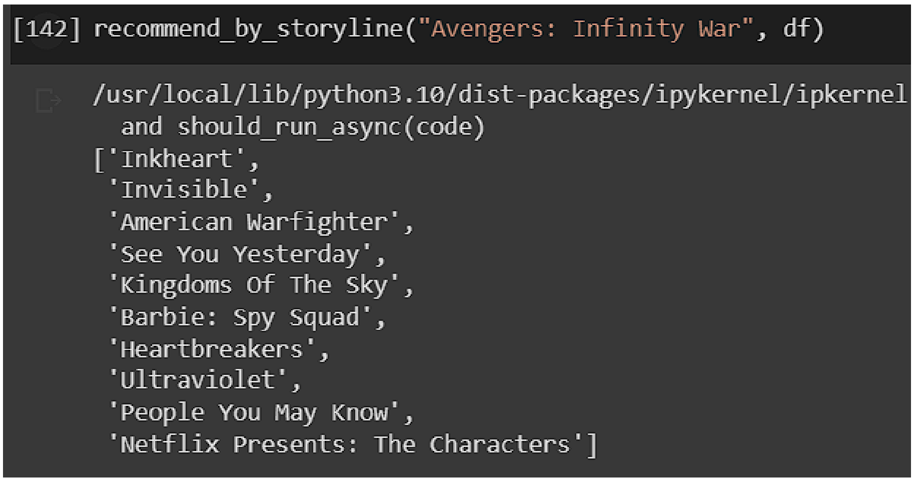
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Figure 5.4: Intertopic Distance Maping for the model

The bar chart changes to show the most important terms found in the chosen topic when you choose a topic from the intertopic distance map or specify a topic in the top panel. The frequency of words that are associated with the chosen topic is shown in a second, darker bar above the term's overall frequency. If the dark bar completely covers the light bar, the word almost solely refers to the chosen subject.

For the recommendation system, the title of the given movie will be taken into consideration. By using the LDA model, we will extract the most probable topic in the given document. We will sort the other documents appearing in this topic distribution in descending order so that we could get most probable document in the selected topic. Then in this sorted data, we will extract the 5 titles above the given movie title and 5 titles below it.

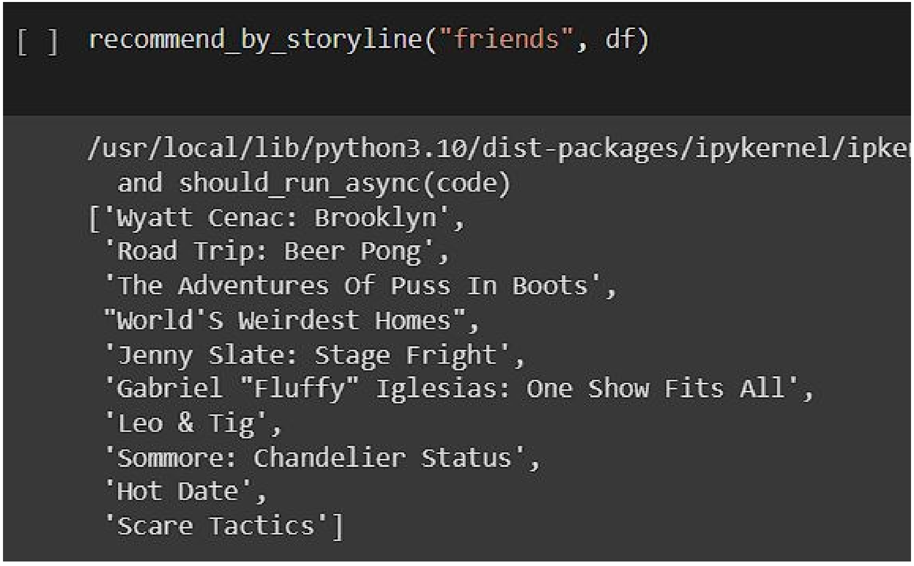
From the results in the figures 5.5 and 5.6, we can see that in "Avengers" input title we got movie recommendation like "American Fighter" and "inkHeart" which are thrillers and fantasy/adventure like the given input. Similarly in the second input when we gave input as "Friends" which is a famous American Sit-com series, we got recommendation of some other comedy shows and documentaries.



```
[142] recommend_by_storyline("Avengers: Infinity War", df)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipkernel
  and should_run_async(code)
['Inkheart',
 'Invisible',
 'American Warfighter',
 'See You Yesterday',
 'Kingdoms Of The Sky',
 'Barbie: Spy Squad',
 'Heartbreakers',
 'Ultraviolet',
 'People You May Know',
 'Netflix Presents: The Characters']
```

Figure 5.5: Movie recommendation based on ”Avengers”



```
[ ]  recommend_by_storyline("friends", df)

/usr/local/lib/python3.10/dist-packages/ipykernel/ipke
  and should_run_async(code)
['Wyatt Cenac: Brooklyn',
 'Road Trip: Beer Pong',
 'The Adventures Of Puss In Boots',
 "World'S Weirdest Homes",
 'Jenny Slate: Stage Fright',
 'Gabriel "Fluffy" Iglesias: One Show Fits All',
 'Leo & Tig',
 'Sommore: Chandelier Status',
 'Hot Date',
 'Scare Tactics']
```

Figure 5.6: Movie recommendation based on ”Friends”

CHAPTER 6

Conclusions & Scope for Future Work

Preface

Based on the analysis and results provided, the objectives of this thesis are summarised in the last chapter. Additionally, it critically analyses the topics that are not covered in the current work and identifies those that require additional research in order to advance the body of knowledge in the field of Natural Language Processing and more.

6.1 Conclusions

In computer science, topic models are crucial for text mining. A topic in topic modelling is a collection of words that appear in statistically significant ways. A text can be any type of unstructured text, including emails, book chapters, blog posts, journal articles, and blog posts. Topic models are unable to comprehend the meanings and concepts of words in text documents. Instead, they make the assumption that any portion of the text is put together by choosing terms from likely word baskets, where each basket corresponds to a topic. The technology repeats this process until it settles on the most likely distribution of words into topic-specific baskets. In terms of the collection as a whole, the individual documents, and the relationships between the documents, topic modelling can offer a helpful view of a huge collection. In this paper we discussed how Latent Dirichlet Allocation works and how this is used to carry out the task of topic modelling. LDA is an unsupervised machine learning technique that works basically on the Dirichlet distributions. The LDA model is a generative model, that is, it can generate a bunch of documents by using its dirichlet distributions of words over topics and topics over documents. In layman words, we backtracked this process to find out the right parameters for a LDA model to generate the original given set of documents. Or we can say that we tried to maximize the probability of documents that were generated were close to the original documents. Gibbs Sampling was used to carry out this inference, train our LDA model and finally formulate the hyper parameters of the LDA model. By doing so, we got the words over topic distributions and topics over document distributions to finally get the topics. As the process is unsupervised and we do not know the optimal number of topics to begin with, we used the coherence score to get the optimal number of topics.

6.2 Contributions Made by the Scholar

(i) Latent Dirichlet Allocation research papers were investigated and gathered to understand the principle of dirichlet distributions and how they perform the task of topic modelling.

(ii) The LDA model was built on an unseen dataset of Netflix Movies and TV shows and later the latent or hidden topics among the dataset were extracted by identifying the underlying statistical inference between documents.

(iii) The topic modelling task was expanded further into making a content based recommendation system for Netflix dataset.

6.3 Scope for Future Work

Numerous future activities for additional research and experimentation can be done with the use of these data and model foundations.

(i) **Use of BERTopic:**
The generative model of LDA does not make any assumptions about the order of words in document. It treats the dataset as bag of words. BERTopic tends to preserve this contextual relationship. The main benefit of BERTopic over conventional topic models is that there is almost no pre-processing of the documents necessary prior to modelling: the pre-trained transformer model, sentence-BERT, takes care of identifying the significant portions of the text, so we don't need to do it manually. (But this does not mean it is better over LDA everytime)

(ii) **Strengthen the model through additional features:**
We intend to investigate additional feature combinations for our LDA model, such as interaction terms and edge strength indicators.

(iii) **Hybrid LDA models:**
Models that learn word vector representations have been created in order to solve some of the LDA limitations that have been highlighted. When compared to other hybrid models or the LDA model alone on microblog (i.e., brief) textual data, Yu and Qiu's hybrid model, which extends the user-LDA topic model with the Dirichlet multinomial mixture and a word vector tool, performs well. Another method that is theoretically related can be used with Twitter data. The hierarchical latent Dirichlet allocation (hLDA) uses word2vec, a vector representations approach, to automatically mine the hierarchical dimension of tweets' subjects. In order to create a more useful dimension, it does this by extracting the semantic associations between words in the data.

References

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[2] R. Alghamdi and K. Alfalqi, “A survey of topic modeling in text mining,” *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, vol. 6, no. 1, 2015.

[3] Z. Xu, Y. Liu, J. Xuan, H. Chen, and L. Mei, “Crowdsourcing based social media data analysis of urban emergency events,” *Multimedia Tools Appl.*, vol. 76, p. 11567–11584, may 2017.

[4] M. Hoffman, F. Bach, and D. Blei, “Online learning for latent dirichlet allocation,” in *Advances in Neural Information Processing Systems* (J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds.), vol. 23, Curran Associates, Inc., 2010.

[5] P. Bhattacharya, M. B. Zafar, N. Ganguly, S. Ghosh, and K. P. Gummadi, “Inferring user interests in the twitter social network,” in *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 357–360, 2014.

[6] D. M. Blei, T. L. Griffiths, and M. I. Jordan, “The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies,” *Journal of the ACM (JACM)*, vol. 57, no. 2, pp. 1–30, 2010.

[7] D. M. Blei and J. D. Lafferty, “A correlated topic model of science,” 2007.

[8] Y.-L. Chang and J.-T. Chien, “Latent dirichlet learning for document summarization,” in *2009 IEEE international conference on acoustics, speech and signal processing*, pp. 1689–1692, IEEE, 2009.

[9] R. Das, M. Zaheer, and C. Dyer, “Gaussian lda for topic models with word embeddings,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 795–804, 2015.

[10] X. Cheng, X. Yan, Y. Lan, and J. Guo, “Btm: Topic modeling over short texts,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2928–2941, 2014.

[11] R. Cohen, I. Aviram, M. Elhadad, and N. Elhadad, “Redundancy-aware topic modeling for patient record notes,” *PloS one*, vol. 9, no. 2, p. e87555, 2014.

[12] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, “Fast collapsed gibbs sampling for latent dirichlet allocation,” in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 569–577, 2008.

[13] L. Hagen, “Content analysis of e-petitions with topic modeling: How to train and evaluate lda models?,” *Information Processing & Management*, vol. 54, no. 6, pp. 1292–1307, 2018.

[14] H. Jelodar, Y. Wang, C. Yuan, X. Feng, X. Jiang, Y. Li, and L. Zhao, "Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey," *Multimedia Tools and Applications*, vol. 78, pp. 15169–15211, 2019.

[15] Y. Kim and K. Shim, "Twilite: A recommendation system for twitter using a probabilistic model based on latent dirichlet allocation," *Information Systems*, vol. 42, pp. 59–77, 2014.

[16] J. S. Liu, "The collapsed gibbs sampler in bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 958–966, 1994.

[17] Z. Liu, Y. Zhang, E. Y. Chang, and M. Sun, "Plda+ parallel latent dirichlet allocation with data placement and pipeline processing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–18, 2011.

[18] S. Syed and M. Spruit, "Full-text or abstract? examining topic coherence scores using latent dirichlet allocation," pp. 165–174, 10 2017.

[19] V. Rus, N. Niraula, and R. Banjade, "Similarity measures based on latent dirichlet allocation," in *Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I 14*, pp. 459–470, Springer, 2013.

[20] C. E. Moody, "Mixing dirichlet topic models and word embeddings to make lda2vec," *arXiv preprint arXiv:1605.02019*, 2016.

[21] D. Newman, P. Smyth, M. Welling, and A. Asuncion, "Distributed inference for latent dirichlet allocation," *Advances in neural information processing systems*, vol. 20, 2007.

[22] M. J. Paul and M. Dredze, "A model for mining public health topics from twitter," *Health*, vol. 11, no. 16-16, p. 1, 2012.

[23] J. Petterson, W. Buntine, S. Narayanamurthy, T. Caetano, and A. Smola, "Word features for latent dirichlet allocation," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[24] X. Sun, B. Li, H. Leung, B. Li, and Y. Li, "Msr4sm: Using topic models to effectively mining software repositories for software maintenance tasks," *Information and Software Technology*, vol. 66, pp. 1–12, 2015.

Index

Author's Biography

Vipul Sangode was born in Chhatisgarh, India on 24th December, 2000. He is currently pursuing B.Tech in Computer Science and Engineering from Dr SPM International Institute of Informational Technology Naya Raipur and is in his final semester. He will be recieving his Bachelor's degree in July 2023. His research focuses on fields of Natural Language Processing and Machine Learning. He aspires to be a Data Scientist and contribute for the nation. He can be contacted at: vipulsangode@gmail.com & vipul19100@iiitnr.edu.in.