“Indexing Contents of Documents Using LDA”

Jaivant Vassan **19BCE2322**

Kabhilan S **19BCE2339**

Prithvi Saran S **19BCE2344**

Submitted to

Prof. Geraldine Bessie Amali, SCOPE

School of Computer Science and Engineering

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Contents

Abstract

Our project 'Indexing Contents of Documents Using LDA' tackles the problem of inefficient search results in the case of localized search engines. Companies, industries, websites, educational institutions etc. house a large amount of textual data which is impossible to classify and index according to topics. Furthermore, search results on such documents return only the most keyword heavy result whereas in some cases the user would require related topics to a certain document. This may be achieved through the use of topic modelling wherein the relationship between words and phrases are analysed algorithmically. There are several algorithms which can help achieve this, but the most accurate and efficient algorithm is LDA, which we have used in our implementation. LDA is a topic modelling algorithm that stands for Latent Dirichlet Allocation. The goal of LDA is to learn the representation of a fixed number of topics, as well as the topic distribution of each document in a collection of documents. In our approach we have focused on the case of research documents. Most specific and non-specific online libraries for research papers are designed to return keyword-based results. In our project we have used a dataset containing research papers from multiple topics and we aim to use the LDA model to determine the most probabilistically related documents to an unseen document. The model achieves this by finding the hidden relationships between the documents that would not be identifiable by traditional search algorithms. With this we've been able to identify related documents with a very high accuracy as well as accurately classify a document to certain topics in the process. We have implemented an interface through which any unseen document can be run through the model which then presents the most related documents to the user. The number of relevant results would only increase with a larger dataset. Thus, this shows that this may be applied to specific cases in different industries with good results.

1. Introduction

The topic for our project is ‘Indexing Contents of Documents Using LDA’. Topic modelling gives us a method to organize, comprehend and summarize huge amounts of data in the form of text. It aids in the discovery of hidden subject patterns in a collection of documents, the annotation of documents according to these topics, and the use of these annotations to organize, search, and summarize texts. Topic modelling is a technique for extracting a group of words or a topic from a set of documents that represent the information contained in the set in the best way possible. It's also a type of text mining which is a technique for identifying reoccurring patterns of words in textual data. To obtain topic models, a variety of methods are employed. We've gone with Latent Dirichlet Allocation (LDA), which is a popular subject modelling technique. Through our literature survey of the LDA topic modelling technique, we have observed that it is highly accurate in discovering the hidden topics or relations between documents. Each document in the LDA model is considered as a mash-up of subjects found in the corpus. According to the approach, each word in the document corresponds to one of the document's subjects. The LDA model identifies the many subjects represented by the documents, as well as how much of each topic is included in each document.

Our aim with this project is to provide a better method of obtaining related documents than an ordinary search engine in the sense of a particular industry or trope of study with the use of the aforementioned topic modelling technique. Traditional search engines used a ranking mechanism that rewarded pages based on how closely the terms in the query matched the words on the page. The more closely the query phrase matches a phrase that appears frequently on your web page, the more likely your web page will be supplied as a relevant result to a searcher.

With the vast amount of textual data present in today’s world it is becoming increasingly necessary to use search functions to find the data we require. While in some specific cases keyword-based search engines are exactly what is required, in other cases search functions are severely lacking the ability to find search results through the relation of different topics. Some major areas where such an approach would have major benefits in indexing, grouping, classifying and finding related documents are in the research field, internal search engines of organizations, industries, schools and universities, internal search engines of websites etc. For our project we have taken the case of the research field. There are numerous websites pertaining to research papers as a whole, as well as specific fields of research which hold vast swathes of documents. Endless amounts of research on various topics are added by the minute and most of these are unclassified documents. It would not be humanly possible to categorize and index such amounts of data without an accurate software. Another problem is the difficulty of finding related and existing research papers on a certain topic or field of research when there is a lack of keywords present between the titles of the documents. Our project when applied to this case and with the necessary data aims to classify the documents by related topics thus providing an easy means of indexing and searching for documents.

2. Hardware/Software Requirements

Hardware:

- Operating system: Linux- Ubuntu 16.04 to 17.10, or Windows 7 to 10
- 2GB RAM (4GB preferable)

Software:

- Python [pandas, numpy and sklearn packages]
- Ngrok

3. Existing System/Approach/Method

[1] Tweeting on COVID-19 pandemic in South Africa: LDA-based topic modelling approach 2020

This paper is mostly focused on the topics that are being discussed on social media during the pandemic period. They discovered the mots talked subjects using the Latent Dirichlet Allocation (LDA) technique. Because twitter was the most popular app during the pandemic, the data was pulled specifically from it. Alcohol sales and consumption, remaining at home, daily statistics tracing, police brutality, 5G, and vaccination conspiracy theories were all heavily discussed on Twitter. They also emphasize the economic downturn brought on by the pandemic. The LDA algorithm aids in the extraction of the most stereotypical subjects. This study was mainly aimed at finding out the most topical issues relating to the COVID-19 pandemic that are being discussed by South Africans, the LDA algorithm was able to bring out related headlines. They are also planning to test other topics involved with natural language processing and machine learning to predict diverse factors relating to the pandemic.

[2] Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques
2020

In this paper they decide to use Agriculture Named Entity Recognition using Topic Modelling techniques. This research employs a topic modeling approach based on Latent Dirichlet Allocation (LDA) to detect soil types, crop diseases, and fertilizers. Because doing all of this work by hand takes a long time, they've tried an unstructured way to identifying the agricultural features, which makes the job go faster. They evaluated roughly 3000 words to recognize names of crops, soil types, fertilizers, and other agricultural terms in order to create a document on agriculture. The terms discovered can be used to build a knowledge base in the agriculture domain that can be used in our applications. They also intend to look at subject clusters and relationships between various entities.

[3] Latent Dirichlet allocation (LDA) and topic modelling: models, applications, a survey 2018

This research will be highly useful and valuable for introducing LDA techniques in topic modelling, according to earlier work. They studied highly academic works (from 2003 to 2016) linked to topic modelling based on LDA in order to uncover the research progress, current trends, and intellectual structure of topic modelling in this paper. They also outlined obstacles and introduced well-known tools and datasets in LDA-based topic modelling.

[4] Understanding Individualization Driving States via Latent Dirichlet Allocation Model 2019

The goal of this research is to create an unsupervised method for learning more about individualization driving. To begin, an encoding approach for extracting driving behaviour from vehicle motion data is proposed. Then, utilizing driving behaviours, a novel Latent Dirichlet Allocation (LDA) model is constructed to identify latent driving states and quantitative structure of driving behaviour patterns (themes) from individualization driving (documents) (words). Twenty-two drivers (15 males and 7 females) were recruited to conduct road trials in Wuhan, China, in order to validate the suggested method's performance and effectiveness. In addition, we create two common unsupervised approaches, k-means and the random method, and compare their performance in our tests. The proposed method's advantage over existing methods is supported by experimental data.

[5] Full-Text or Abstract? Examining Topic Coherence Scores Using Latent Dirichlet Allocation, 2017

This study assesses topic coherence and human topic assessment of revealed latent subjects from scientific publications using the topic model latent Dirichlet allocation (LDA) on abstract and full-text data. The distributional hypothesis states that words with similar meanings are more likely to appear in the same context, and subject coherence is used as a proxy for topic quality. Although machine-learning researchers have given LDA a lot of attention, especially with its modifications and extensions, little is known about the effects of different types of textual data on generated themes. This is the first study to look into these practical impacts, and it reveals that document frequency, document word length, and vocabulary size all have mixed effects on topic coherence and human topic ranking of LDA themes. They also show that inaccurate or noisy terms in topic-word distributions have less of an impact on big document collections, causing subjects to be more coherent and rated higher. Within small document collections, discrepancies between abstract and full-text data are more noticeable, with differences as great as 90% high-quality themes for full-text data versus 50% high-quality subjects for abstract data.

[6] Topic modelling discourse dynamics in historical newspapers, 2020

This paper focuses on historical researches. They are using two topic models that is DTM and LDA on discovering dynamics on old newspapers. Their main conclusion is to do a combined sampling, training and inference course of action on a huge synchronic text, they also do a discussion on different topic models for this data. They're using papers from the nineteenth century that were made in Finland. They propose an approach in this research by selecting various algorithms for analytical guides and possible outcomes. The LDA and DTM topic models can quickly detect comparable subjects in many newspapers from the time period. The LDA is utilized as a tool for interpreting issues that are even vaguely connected. They also state that utilizing LDA is far superior to using the DTM topic model, especially when it comes to trustworthy quantification. In this publication, they make no mention of their future projects.

[7] Discovering Trends of Mobile Learning Research Using Topic Modelling Approach 2020

This article examines how mobile learning has evolved over the last ten years. They also arrive at the conclusion that students prefer mobile learning. They describe topic modelling, and the Latent Dirichlet Allocation is the most commonly used variant. The paper is divided into five sections, each of which describes the author's work. The fact that mobile learning is the most well-known topic of research has caused people to grasp the significance of this. According to topic modelling, there are numerous topics that have been identified as prominent themes in the literature over the last ten years.

[8] Analysing the Social-Economic Impact of Wireless Mobile Services During and Before COVID-19 Using Topic Modelling and Sentiment Analysis on Tweets
2021

Using Topic Modelling and Sentiment Analysis on Tweets, we examined the social-economic impact of wireless mobile services during and before COVID-19. Wehel Hadi compares three models of analysis by using them to analyse the influence of wireless mobile services on customer feelings such as user enjoyment (social effect), affordability (economic effect), and desire to pay (willingness to pay) (social effect). The test environment is Twitter, where user reactions are similar to those seen in a typical social situation. This research is also based on two distinct mobile phone companies with varied policy flexibility to see if a company's policy effects the sentiment of fundamental values such user happiness (social effect), affordability (economic effect), and desire to pay (willingness to pay) (social effect). Finally, there is a bad customer sentiment, which is heightened owing to the epidemic. In compared to a company with more traditional policies, a mobile company with a more flexible policy has a lower negative rate. LDA proves to be a great model with 98 percent accuracy and the highest coherence rate of 0.55 in comparison to LSA with 0.52 and HDP with 0.19 in an investigation utilizing the software PyLDAvis using various classifiers and being optimized through hyper parameter tweaking, over four social subjects.

3.1 Drawbacks of existing System

Traditionally documents were indexing, search engines and search functions used keywords. However, a rigid keyword-based ranking method has drawbacks. For starters, keyword spam is a simple way to game this system. And, perhaps more importantly, searchers are shifting away from keyword phrases in favour of plain language inquiries. The popularity of mobile Internet use and queries shouted into cell phones via voice assistants is a major factor in this trend. Search engines created technology that understands the connection between words through keyword relationships in order to offer the best results for queries in natural language format. However, the technology driving keyword tools and topic-modelling methodologies differs. Keyword tools typically aggregate together subjects that appear together in high-ranking sites. They don't take into account the number of co-occurrences between terms, semantic importance, or other signals; thus, they can't tell you how closely two topics are related. Topic modelling, on the other hand, may make this computation and rank related subjects by relevance while also taking into consideration other on-page factors. Another major problem is that though major web page search engines from large companies like Google and Microsoft employ AI and NLP based approaches to find relevant web pages, more localized search engines and software are left behind.

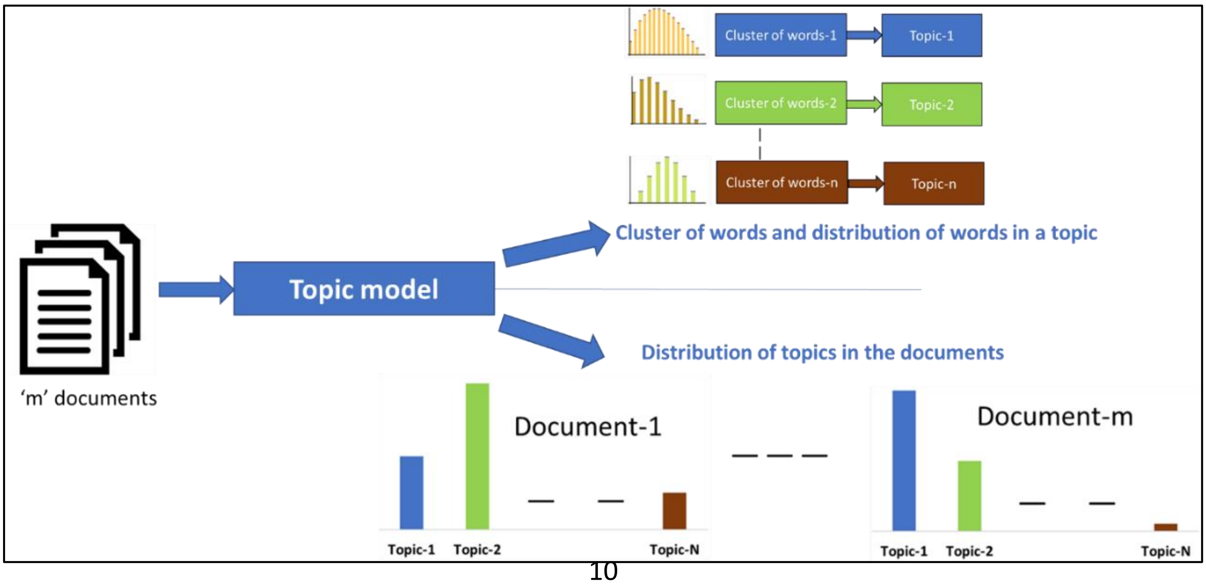
4. Proposed/Developed Model

Topic Modelling:

Topic modeling is an unsupervised machine learning technique that can scan a series of documents, find word and phrase patterns within them, and automatically cluster word groups and related expressions that best characterize the set. Because it doesn't require a preexisting list of tags or training data that has been previously classified by humans, this is referred to as 'unsupervised' machine learning. Topic modeling is a quick and straightforward technique to start examining data because it doesn't require any training. To infer subjects from unstructured data, topic modeling involves counting words and grouping similar word patterns. A topic model clusters feedback that is comparable, as well as phrases and expressions that appear most frequently, by detecting patterns such as word frequency and distance between words. You may rapidly deduce what each group of texts is about using this information. Remember that this method is 'unsupervised,' which means that no prior training is required. You'll get collections of documents that the algorithm has grouped together, as well as clusters of words and expressions that it used to infer these relationships, at the end of your topic modeling investigation. The method of separating a corpus of texts into two parts is known as topic modeling.

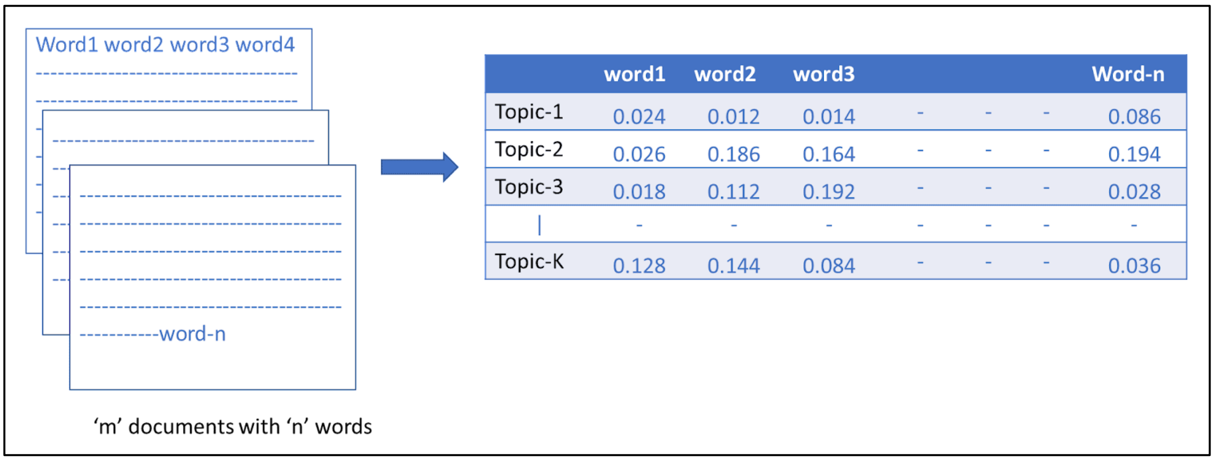
1. A list of the subjects covered by the corpus documents.
2. A collection of papers from the corpus categorized by the subjects they address.

Every document contains a statistical mixture of subjects, i.e., a statistical distribution of topics that can be generated by "adding up" all of the distributions for all of the topics mentioned. Topic modelling methods attempt to determine which subjects are present in the corpus' documents and how strong their presence is.



Latent Dirichlet Allocation (LDA):

The term 'latent' refers to the model's discovery of 'yet-to-be-discovered' or concealed subjects in documents. The premise of LDA is that the distribution of themes in a document and the distribution of words in subjects are both Dirichlet distributions. The term 'allocation' refers to how the document's themes are distributed. LDA considers that documents are made up of words that help determine the subjects, and it maps texts to a topic list by allocating each word to a different topic. The figure shows the assignment in terms of conditional probability estimates. The probability of a word 'wj' relating to subject 'tk' is represented by the value in each cell in the figure. The word and topic indices are denoted by 'j' and 'k,' respectively. It's worth noting that LDA ignores the sequence in which words appear and syntactic information. It considers documents to be nothing more than a collection of words or a bag of words.



	word1	word2	word3				Word-n
Topic-1	0.024	0.012	0.014	-	-	-	0.086
Topic-2	0.026	0.186	0.164	-	-	-	0.194
Topic-3	0.018	0.112	0.192	-	-	-	0.028
|	-	-	-	-	-	-	-
Topic-K	0.128	0.144	0.084	-	-	-	0.036

Once the probabilities are estimated, finding the collection of words that reflect a specific topic can be done either by picking the top 'r' probabilities of words or by setting a probability threshold and selecting only the words whose probabilities are greater than or equal to the threshold value. For example, if we concentrate on topic-1 in the figure and select the top four probabilities, assuming that the probabilities of the words not listed in the table are less than 0.012, topic-1 can be expressed using the 'r' top probabilities words approach as shown below.

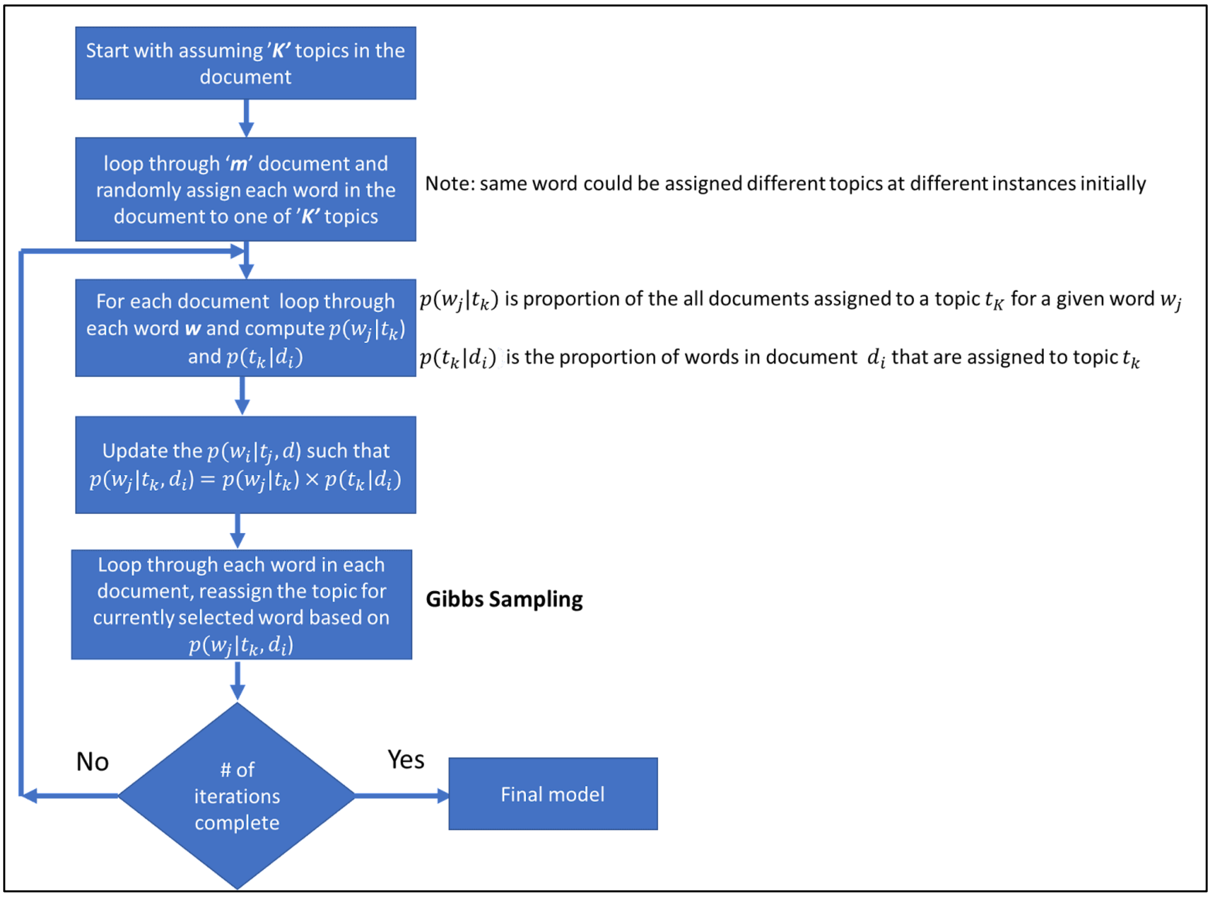
If word-k, word1, word3, and word2 are trees, mountains, rivers, and streams, then topic-1 may be 'nature' in the example above.

The number of expected topics in the texts is a significant input to LDA. If we set the expected topics to 3 in the previous example, each document can be represented as shown below.

$$D_i = w_{1_i} \times Topic - 1 + w_{2_i} \times Topic - 2 + w_{3_i} \times Tpoic - 3$$

LDA Algorithm:

Each document is generated via a statistical generating process, according to LDA. That is, each document is made up of a variety of topics, each of which is made up of a variety of words. LDA reverses the generating process while recognising the subjects in the documents. The flowchart below depicts the general steps involved in the procedure. It's worth noting that LDA starts with a random assignment of subjects to each word and then improves the assignment of topics to words iteratively using Gibbs sampling.


Start with assuming '***K'*** topics in the document
loop through '***m'*** document and randomly assign each word in the document to one of '***K'*** topics
Note: same word could be assigned different topics at different instances initially
For each document loop through each word ***w*** and compute $p(w_j|t_k)$ and $p(t_k|d_i)$
$p(w_j|t_k)$ is proportion of the all documents assigned to a topic t_K for a given word w_j
$p(t_k|d_i)$ is the proportion of words in document d_i that are assigned to topic t_k
Update the $p(w_i|t_j, d)$ such that $p(w_j|t_k, d_i) = p(w_j|t_k) \times p(t_k|d_i)$
Loop through each word in each document, reassign the topic for currently selected word based on $p(w_j|t_k, d_i)$
Gibbs Sampling
# of iterations complete
No
Yes
Final model

Hyper Parameters in LDA:

There are three hyper parameters in LDA:

1) 'α' denotes the document to topic density factor

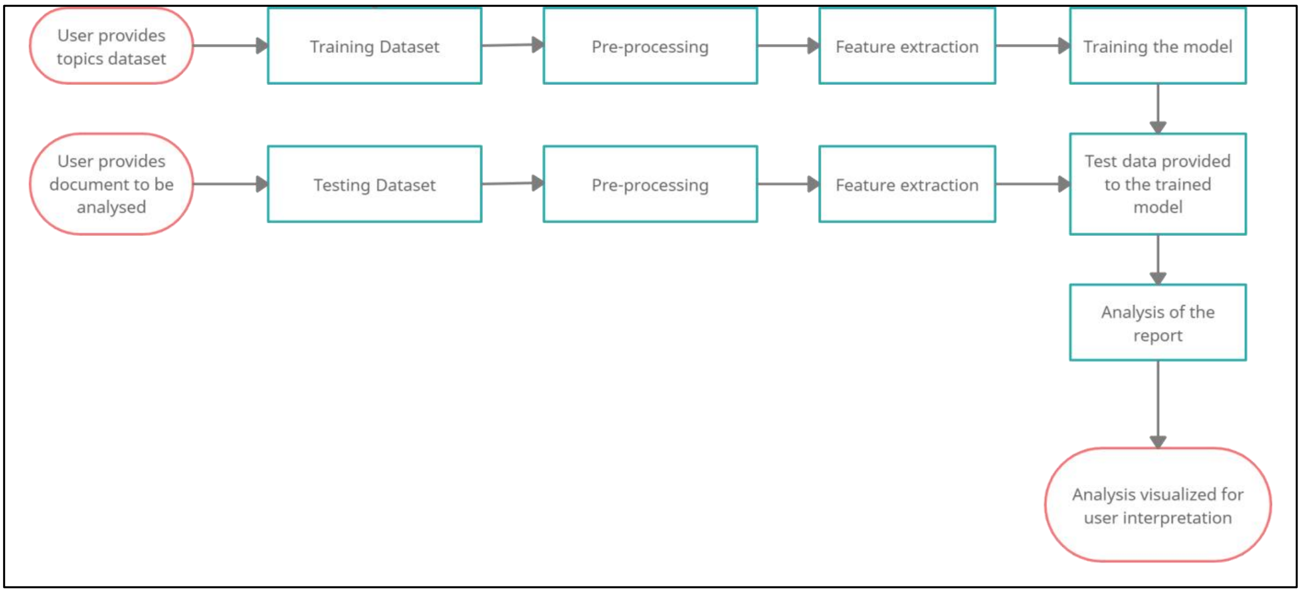
2) 'β' denotes the topic to word density factor

3) 'K' denotes the number of topics

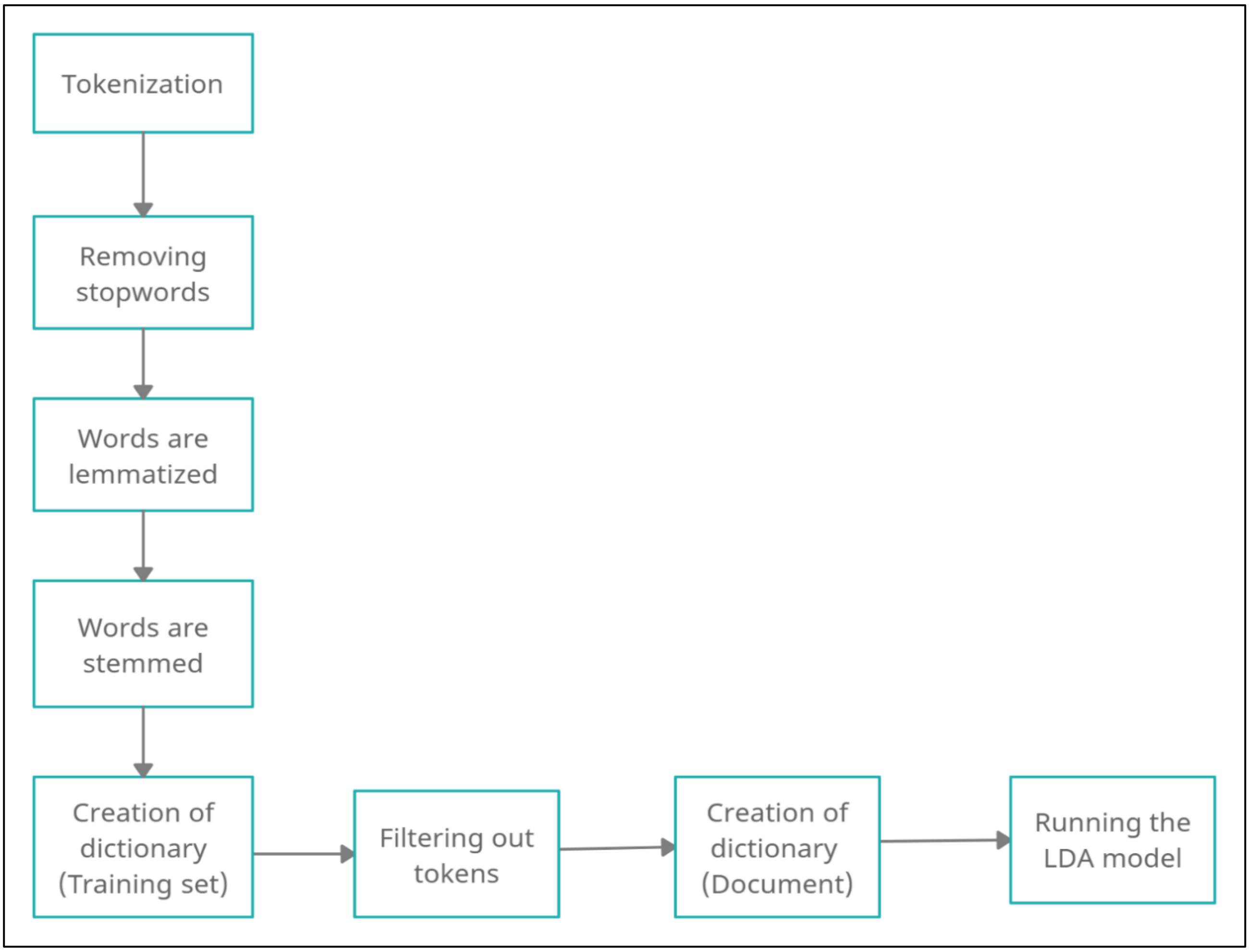
- The ‘α’ hyperparameter determines how many subjects should be included in the document. A low value of ‘α’ indicates that the papers should have fewer themes in the mix & greater value indicates that the documents should have more topics in the mix.
- The ‘β’ hyper parameter determines how many words are distributed each topic. Topics with lower values of ‘β’ will generally have fewer words, whereas topics with higher values will likely have more words.
- In an ideal world, each document would have a few topics and a few words for each of the topics. As a result, ‘α’ and ‘β’ are usually set below one.
- The 'K' hyperparameter defines the expected number of themes in the document corpus. K is usually assigned a value based on domain knowledge.

4.1 Design

Architecture Diagram:


User provides topics dataset
Training Dataset
Pre-processing
Feature extraction
Training the model
User provides document to be analysed
Testing Dataset
Pre-processing
Feature extraction
Test data provided to the trained model
Analysis of the report
Analysis visualized for user interpretation

Proposed Model Diagram:



4.2 Module Wise Description

Pre-processing:

The first module in our proposed model is pre-processing. We perform the following processes to pre-process the data:

- Splitting the text to sentences and also the sentences to words using tokenization.
- Removal of all punctuation and conversion of the words to lower case.
- Omission of words with less than 3 characters.
- Elimination of stop words.
- Transformation of words in the third person to the first person and verbs in the past and future tense to the present tense.
- Words are stemmed, which means they are reduced to their simplest form.

Bag of Words:

In the next module we create a dictionary from the processed dataset containing the number of times a word appears in the training set. This is known as the bag of words model. A bag-of-words model extracts features from text for modelling purposes. The method is straightforward and adaptable, and it may be used to extract information from documents in a variety of ways. A bag-of-words is a text representation that describes the frequency with which words appear in a document. It has two major properties:

1. A list of terms that are well-known.
2. A metric for determining the existence of well-known terms.

Because any information about the sequence or structure of words in the document is deleted, it is referred to as a "bag" of words. The model simply cares about whether or not recognized terms appear in the document, not where they appear.

We then filter out tokens from the bag of words based on certain parameters

- Tokens appearing in less than 15 documents (absolute number) are removed.
- Tokens appearing in more than 0.5 documents (fraction of corpus size) are removed.
- Only the first 50,000 most frequent tokens are kept.

We then generate a dictionary for each document that lists the number of words and how often they appear.

TF-IDF:

Next, we create we create our TF-IDF model using the bag of words corpus and apply transformations on the entire corpus. The TF-IDF (term frequency-inverse document frequency) statistic examines the relevance of a word to a document in a collection of documents. The TF-IDF format was created for document search and retrieval. It works by growing in proportion to the number of times a word appears in a document, but offset by the number of papers containing the word.

For each word in a document, the TF-IDF is calculated by multiplying two metrics:

- The term for the number of times a word appears in a document. The simplest method for calculating this frequency is to simply count the number of times a word appears in a document. The frequency can then be adjusted based on the length of the document or the raw frequency of the most frequently used word in the document.
- The word's inverse document frequency over a collection of documents. This refers to how common or uncommon a word is within the entire document set. The closer a term is to zero, the more common it is.
- This number will approach 0 if the word is exceedingly common and appears in numerous documents. Otherwise, it will be close to 1.

The TF-IDF score of a word in a document is calculated by multiplying these two integers. The greater the score, the more important the word in that paper is.

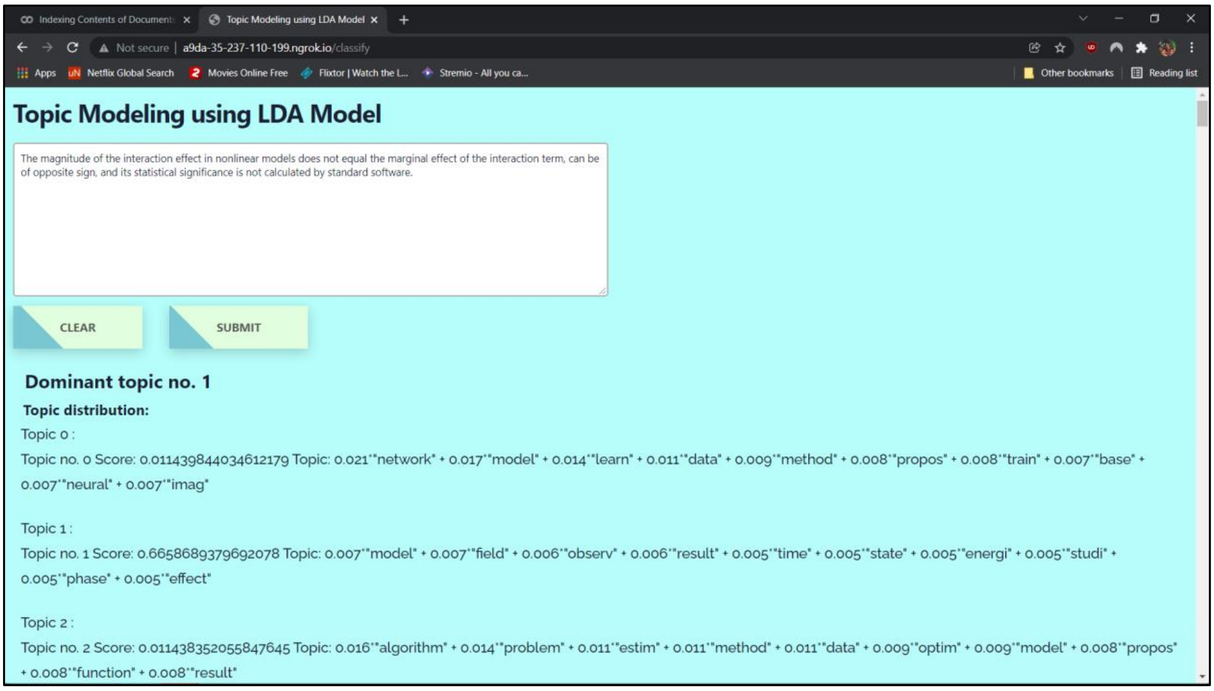
LDA Model:

We then run the LDA model using Bag of Words as well as TF-IDF. We get the terms that appear in each topic, as well as their proportional weight, for each topic.

Frontend Interface:

We have implemented a front-end webpage to obtain the input from the user as an unseen document. The unseen document is then processed in the same way as the training data and using either of the two models we evaluate to which topics and document the unseen document has a higher probability of being classified and related to respectively. The top 5 most dominant topic and the topic distribution of the resulting documents are displayed to the user along with the most relevant documents to the unseen documents having the highest accuracy.

4.3. Implementation



Topic Modeling using LDA Model

The magnitude of the interaction effect in nonlinear models does not equal the marginal effect of the interaction term, can be of opposite sign, and its statistical significance is not calculated by standard software.

CLEAR

SUBMIT

Dominant topic no. 1

Topic distribution:

Topic 0 :

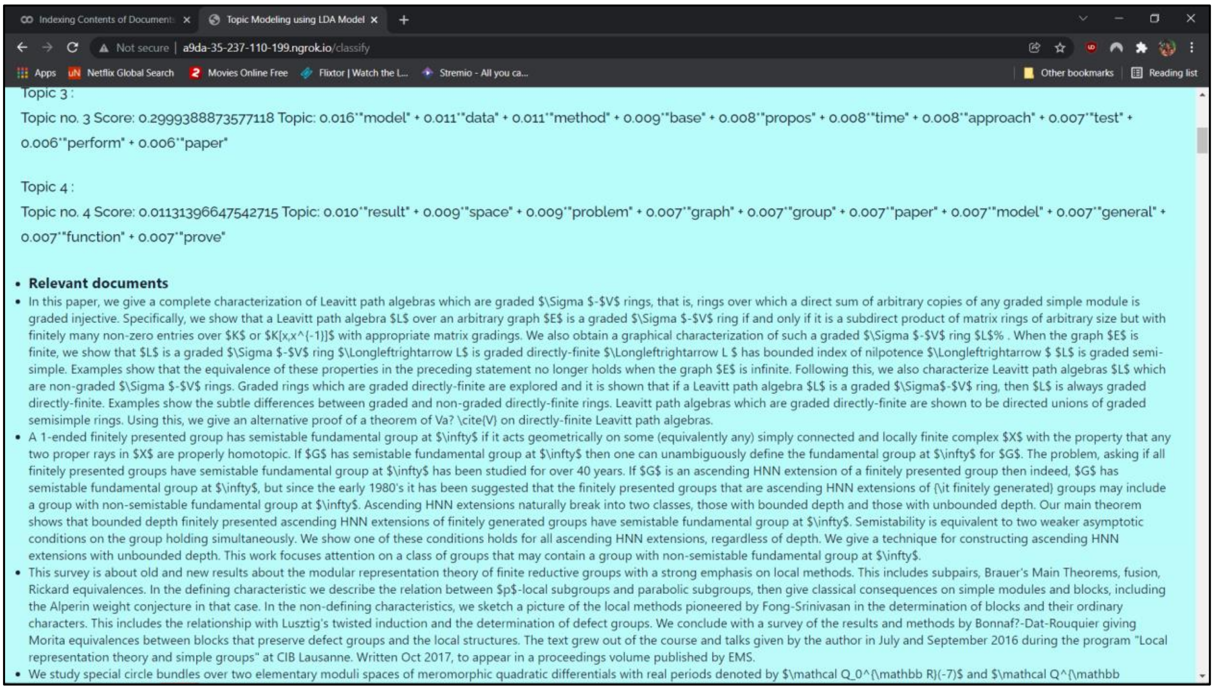
Topic no. 0 Score: 0.011439844034612179 Topic: 0.021*"network" + 0.017*"model" + 0.014*"learn" + 0.011*"data" + 0.009*"method" + 0.008*"propos" + 0.008*"train" + 0.007*"base" + 0.007*"neural" + 0.007*"imag"

Topic 1 :

Topic no. 1 Score: 0.6658689379692078 Topic: 0.007*"model" + 0.007*"field" + 0.006*"observ" + 0.006*"result" + 0.005*"time" + 0.005*"state" + 0.005*"energi" + 0.005*"studi" + 0.005*"phase" + 0.005*"effect"

Topic 2 :

Topic no. 2 Score: 0.011438352055847645 Topic: 0.016*"algorithm" + 0.014*"problem" + 0.011*"estim" + 0.011*"method" + 0.011*"data" + 0.009*"optim" + 0.009*"model" + 0.008*"propos" + 0.008*"function" + 0.008*"result"



Topic 3 :

Topic no. 3 Score: 0.2999388873577118 Topic: 0.016*"model" + 0.011*"data" + 0.011*"method" + 0.009*"base" + 0.008*"propos" + 0.008*"time" + 0.008*"approach" + 0.007*"test" + 0.006*"perform" + 0.006*"paper"

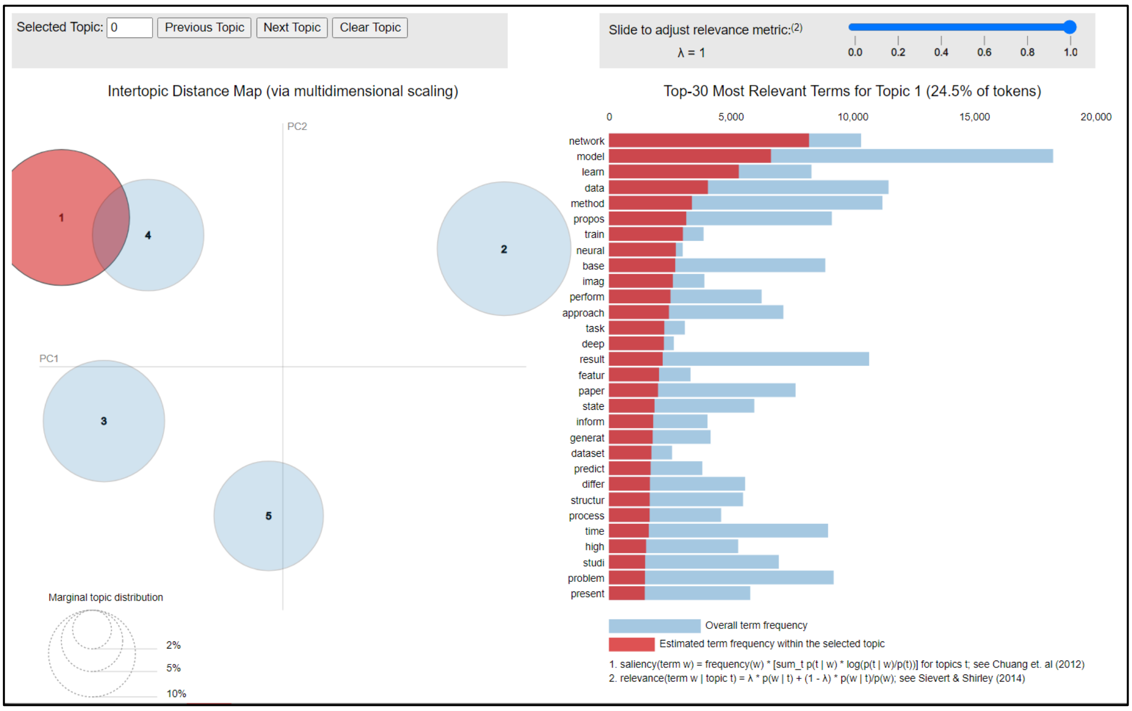
Topic 4 :

Topic no. 4 Score: 0.01131396647542715 Topic: 0.010*"result" + 0.009*"space" + 0.009*"problem" + 0.007*"graph" + 0.007*"group" + 0.007*"paper" + 0.007*"model" + 0.007*"general" + 0.007*"function" + 0.007*"prove"

- **Relevant documents**
- In this paper, we give a complete characterization of Leavitt path algebras which are graded $\Sigma $-$V$ rings, that is, rings over which a direct sum of arbitrary copies of any graded simple module is graded injective. Specifically, we show that a Leavitt path algebra L over an arbitrary graph E is a graded $\Sigma $-$V$ ring if and only if it is a subdirect product of matrix rings of arbitrary size but with finitely many non-zero entries over K or $K[x,x^{-1}]$ with appropriate matrix gradings. We also obtain a graphical characterization of such a graded $\Sigma $-$V$ ring L% . When the graph E is finite, we show that L is a graded $\Sigma $-$V$ ring $\Longleftrightarrow L$ is graded directly-finite $\Longleftrightarrow L $ has bounded index of nilpotence $\Longleftrightarrow $ L is graded semi-simple. Examples show that the equivalence of these properties in the preceding statement no longer holds when the graph E is infinite. Following this, we also characterize Leavitt path algebras L which are non-graded $\Sigma $-$V$ rings. Graded rings which are graded directly-finite are explored and it is shown that if a Leavitt path algebra L is a graded Σ-V ring, then L is always graded directly-finite. Examples show the subtle differences between graded and non-graded directly-finite rings. Leavitt path algebras which are graded directly-finite are shown to be directed unions of graded semisimple rings. Using this, we give an alternative proof of a theorem of Va? \cite{V} on directly-finite Leavitt path algebras.
- A 1-ended finitely presented group has semistable fundamental group at ∞ if it acts geometrically on some (equivalently any) simply connected and locally finite complex X with the property that any two proper rays in X are properly homotopic. If G has semistable fundamental group at ∞ then one can unambiguously define the fundamental group at ∞ for G. The problem, asking if all finitely presented groups have semistable fundamental group at ∞ has been studied for over 40 years. If G is an ascending HNN extension of a finitely presented group then indeed, G has semistable fundamental group at ∞, but since the early 1980's it has been suggested that the finitely presented groups that are ascending HNN extensions of {\it finitely generated} groups may include a group with non-semistable fundamental group at ∞. Ascending HNN extensions naturally break into two classes, those with bounded depth and those with unbounded depth. Our main theorem shows that bounded depth finitely presented ascending HNN extensions of finitely generated groups have semistable fundamental group at ∞. Semistability is equivalent to two weaker asymptotic conditions on the group holding simultaneously. We show one of these conditions holds for all ascending HNN extensions, regardless of depth. We give a technique for constructing ascending HNN extensions with unbounded depth. This work focuses attention on a class of groups that may contain a group with non-semistable fundamental group at ∞.
- This survey is about old and new results about the modular representation theory of finite reductive groups with a strong emphasis on local methods. This includes subpairs, Brauer's Main Theorems, fusion, Rickard equivalences. In the defining characteristic we describe the relation between p-local subgroups and parabolic subgroups, then give classical consequences on simple modules and blocks, including the Alperin weight conjecture in that case. In the non-defining characteristics, we sketch a picture of the local methods pioneered by Fong-Srinivasan in the determination of blocks and their ordinary characters. This includes the relationship with Lusztig's twisted induction and the determination of defect groups. We conclude with a survey of the results and methods by Bonnaf?-Dat-Rouquier giving Morita equivalences between blocks that preserve defect groups and the local structures. The text grew out of the course and talks given by the author in July and September 2016 during the program "Local representation theory and simple groups" at CIB Lausanne. Written Oct 2017, to appear in a proceedings volume published by EMS.
- We study special circle bundles over two elementary moduli spaces of meromorphic quadratic differentials with real periods denoted by $\mathcal Q_0^{\mathbb R}(-7)$ and $\mathcal Q^{\mathbb

5. Results and Discussion

Visualization of Bag of Words Model:



Selected Topic: 0 Previous Topic Next Topic Clear Topic

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

Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Relevant Terms for Topic 1 (24.5% of tokens)

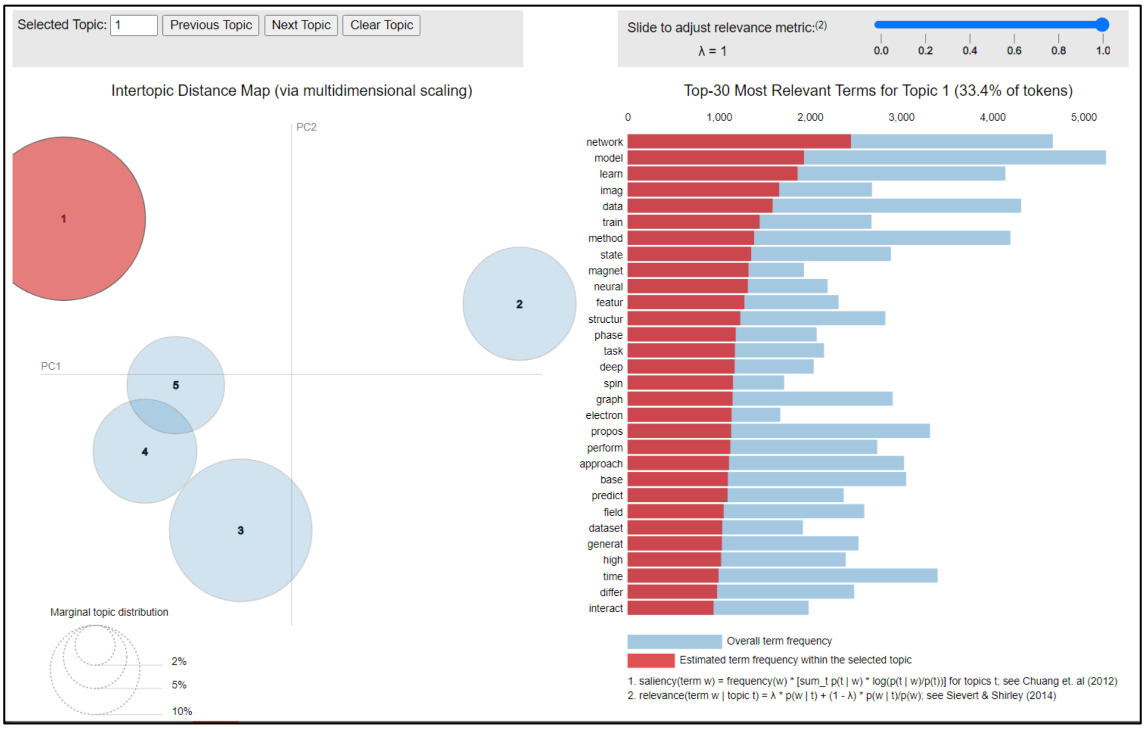
Overall term frequency

Estimated term frequency within the selected topic

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)

We observe that for the Topic-1, in the bag of words model, the most relevant term is ‘network’.

Visualization of TF-IDF Model:



Selected Topic: 1 Previous Topic Next Topic Clear Topic

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

Intertopic Distance Map (via multidimensional scaling)

Top-30 Most Relevant Terms for Topic 1 (33.4% of tokens)

Overall term frequency

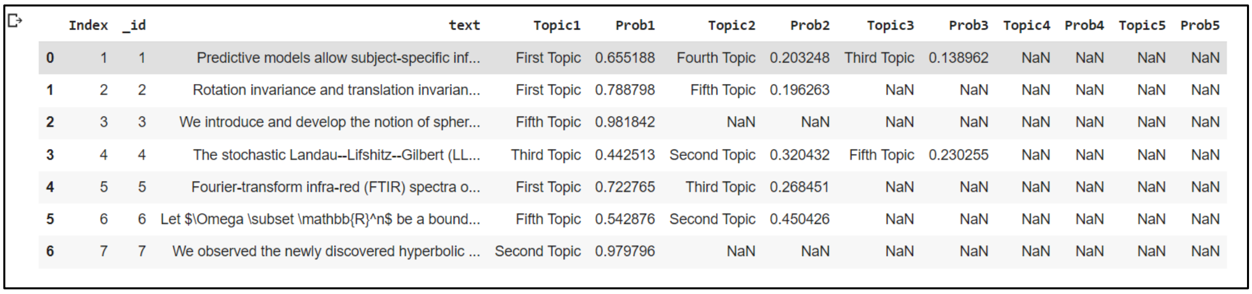
Estimated term frequency within the selected topic

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)

We observe that for the Topic-1, in TF-IDF model, the most relevant term is ‘network’.

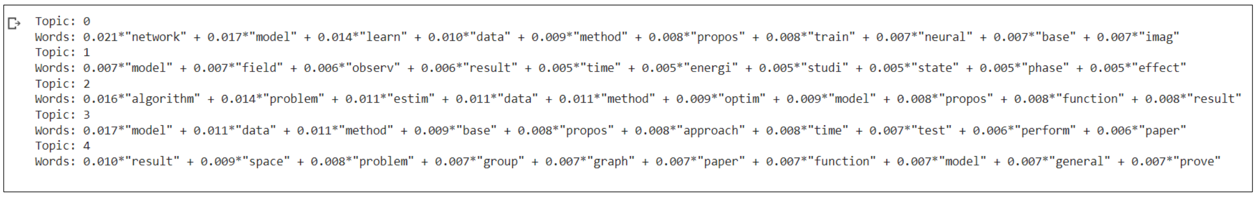
From these visualizations we can conclude that both models are accurate in identifying the most relevant terms for a particular topic since the distributions of relevant terms for multiple topics give similar results which is verifiably accurate.

Documents labelled by topic number in documents data frame:



	Index	_id	text	Topic1	Prob1	Topic2	Prob2	Topic3	Prob3	Topic4	Prob4	Topic5	Prob5
0	1	1	Predictive models allow subject-specific inf...	First Topic	0.655188	Fourth Topic	0.203248	Third Topic	0.138962	NaN	NaN	NaN	NaN
1	2	2	Rotation invariance and translation invarian...	First Topic	0.788798	Fifth Topic	0.196263	NaN	NaN	NaN	NaN	NaN	NaN
2	3	3	We introduce and develop the notion of spher...	Fifth Topic	0.981842	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	4	The stochastic Landau--Lifshitz--Gilbert (LL...	Third Topic	0.442513	Second Topic	0.320432	Fifth Topic	0.230255	NaN	NaN	NaN	NaN
4	5	5	Fourier-transform infra-red (FTIR) spectra o...	First Topic	0.722765	Third Topic	0.268451	NaN	NaN	NaN	NaN	NaN	NaN
5	6	6	Let $\Omega \subset \mathbb{R}^n$ be a bound...	Fifth Topic	0.542876	Second Topic	0.450426	NaN	NaN	NaN	NaN	NaN	NaN
6	7	7	We observed the newly discovered hyperbolic ...	Second Topic	0.979796	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Topic distribution for unseen document:



```
Topic: 0
Words: 0.021*"network" + 0.017*"model" + 0.014*"learn" + 0.010*"data" + 0.009*"method" + 0.008*"propos" + 0.008*"train" + 0.007*"neural" + 0.007*"base" + 0.007*"imag"
Topic: 1
Words: 0.007*"model" + 0.007*"field" + 0.006*"observ" + 0.006*"result" + 0.005*"time" + 0.005*"energi" + 0.005*"studi" + 0.005*"state" + 0.005*"phase" + 0.005*"effect"
Topic: 2
Words: 0.016*"algorithm" + 0.014*"problem" + 0.011*"estim" + 0.011*"data" + 0.011*"method" + 0.009*"optim" + 0.009*"model" + 0.008*"propos" + 0.008*"function" + 0.008*"result"
Topic: 3
Words: 0.017*"model" + 0.011*"data" + 0.011*"method" + 0.009*"base" + 0.008*"propos" + 0.008*"approach" + 0.008*"time" + 0.007*"test" + 0.006*"perform" + 0.006*"paper"
Topic: 4
Words: 0.010*"result" + 0.009*"space" + 0.008*"problem" + 0.007*"group" + 0.007*"graph" + 0.007*"paper" + 0.007*"function" + 0.007*"model" + 0.007*"general" + 0.007*"prove"
```

From this result we can infer that the model has obtained the most relevant result along with the weights of the topics used to classify the document.

6. Conclusion

Through the implementation of our project, ‘Topic Modelling using LDA Model’ we have demonstrated the accuracy of the model in obtaining an accurate topic distribution of related topics as well as related documents to the user’s given document. From these results we conclude that such an approach may be applied to help localized searches in various industries as well as indexing and classification of documents to related topics. This eradicates the requirement of manual work in classifying documents according to topics and highly reduces the difficulty of finding hidden relations in documents when a non-keyword bases approach is applied. Human labour that would have once been needed to check relevance between documents can hence be replaced by such a model.

References

1. Backenroth, D., He, Z., Kiryluk, K., Boeva, V., Pethukova, L., Khurana, E., Christiano, A., Buxbaum, J.D. and Ionita-Laza, I., 2018. FUN-LDA: a latent dirichlet allocation model for predicting tissue-specific functional effects of noncoding variation: methods and applications. *The American Journal of Human Genetics*, *102*(5), pp.920-942.

2. Bastani, K., Namavari, H. and Shaffer, J., 2019. Latent Dirichlet allocation (LDA) for topic modeling of the CFPB consumer complaints. *Expert Systems with Applications*, *127*, pp.256-271.

3. Chen, C., Zare, A., Trinh, H.N., Omotara, G.O., Cobb, J.T. and Lagaunne, T.A., 2017. Partial membership latent Dirichlet allocation for soft image segmentation. *IEEE Transactions on Image Processing*, *26*(12), pp.5590-5602.

4. Chen, Z., Zhang, Y., Wu, C. and Ran, B., 2019. Understanding individualization driving states via latent Dirichlet allocation model. *IEEE Intelligent Transportation Systems Magazine*, *11*(2), pp.41-53.

5. Cheng, X., Cao, Q. and Liao, S.S., 2020. <? covid19?> An overview of literature on COVID-19, MERS and SARS: Using text mining and latent Dirichlet allocation. *Journal of Information Science*, p.0165551520954674.

6. Du, Y., Yi, Y., Li, X., Chen, X., Fan, Y. and Su, F., 2020. Extracting and tracking hot topics of micro-blogs based on improved Latent Dirichlet Allocation. *Engineering Applications of Artificial Intelligence*, *87*, p.103279.

7. Gangadharan, V. and Gupta, D., 2020. Recognizing Named Entities in Agriculture Documents using LDA based Topic Modelling Techniques. *Procedia Computer Science*, *171*, pp.1337-1345.

8. Hadi, W., 2021. *Analysing the Social-Economic Impact of Wireless Mobile Services During and Before COVID-19 Using Topic Modelling and Sentiment Analysis on Tweets* (Master's thesis).
9. Hamzah, A., Hidayatullah, A. and Persada, A., 2020. Discovering trends of mobile learning research using topic modelling approach.

10. Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. and Zhao, L., 2019. Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications*, *78*(11), pp.15169-15211.

11. Jeon, J., Padhye, S., Yoon, S., Cai, H. and Hastak, M., 2021. Identification of metrics for the Purdue index for construction using latent dirichlet allocation. *Journal of Management in Engineering*, *37*(6), p.04021067.

Appendix

```
dictionary = gensim.corpora.Dictionary(processed_docs)

# Saving the dictionary

dictionary.save('dictionary.gensim')

count = 0

for k, v in dictionary.iteritems():

    print(k, v)

    count += 1

    if count > 10:

        break

dictionary.filter_extremes(no_below=15, no_above=0.5, keep_n=50000)

bow_corpus = [dictionary.doc2bow(doc) for doc in processed_docs]

# Saving the corpus

import pickle

pickle.dump(bow_corpus, open('bow_corpus.pkl', 'wb'))

bow_corpus[150]

bow_doc_1500 = bow_corpus[150]

for i in range(len(bow_doc_1500)):

    print("Word {} (\"{}\") appears {} time.".format(bow_doc_1500[i][0],

                                                     dictionary[bow_doc_1500[i][0]],

                                                     bow_doc_1500[i][1]))

from gensim import corpora, models

tfidf = models.TfidfModel(bow_corpus)

corpus_tfidf = tfidf[bow_corpus]

from pprint import pprint
```

```
for doc in corpus_tfidf:
    pprint(doc)
    break
lda_model = gensim.models.LdaMulticore(bow_corpus, num_topics=5,
id2word=dictionary, passes=2, workers=2, random_state= 2)
# Saving the model
lda_model.save('lda_model.gensim')
for idx, topic in lda_model.print_topics(-1):
    print('Topic: {} \nWords: {}'.format(idx, topic))
unseen_document = 'The recent discovery that the exponent of matrix multiplication
is determined by the rank of the symmetrized matrix multiplication tensor has
invigorated interest in better understanding symmetrized matrix multiplication'
bow_vector = dictionary.doc2bow(preprocess(unseen_document))
for index, score in sorted(lda_model[bow_vector], key=lambda tup: -1*tup[1]):
    print("Score: {}\t Topic: {}".format(score, lda_model.print_topic(index, 10)))
from flask import Flask, render_template, request
from flask_ngrok import run_with_ngrok
app = Flask(__name__, template_folder="/content/gdrive/My Drive/templates")
from operator import itemgetter
run_with_ngrok(app)
@app.route("/")
def index():
    a=[1,2,3,4,5]
    return render_template('index.html')
@app.route('/classify',methods=['GET','POST'])
def comparer():
  if request.method == 'POST':
    rawtext = request.form['rawtext']
    unseen_document=rawtext
    bow_vector = dictionary.doc2bow(preprocess(unseen_document))
    a=[]
    a.append(" ")
    dom1=0
    for index, score in lda_model[bow_vector]:
      b=("Topic no. {}\t Score: {}\t Topic: {}".format(index, score,
lda_model.print_topic(index, 10)))
      a.append(b)
  dom1=max(lda_model[bow_vector],key=itemgetter(1))[0]
  return
render_template('index.html',T1=a[1],T2=a[2],T3=a[3],T4=a[4],T5=a[5],dom=dom1,
paras=para())
def para():
  paras=[]
  i=0
  for i in range(10000):
    if db.Prob1[i]>0.99 and db.Topic1[i]==lda_model[bow_vector][dom1][0]:
      paras.append(db.text[i])
  return paras
app.run()
```

****