

Topic Modelling Using Latent Dirichlet Allocation



Submitted by:

Mubeen Khan 2017-EE-14

Shahkar ul Hassan 2017-EE-36

Supervised by: Dr. Kashif Javed

Department of Electrical Engineering
University of Engineering and Technology Lahore

Topic Modelling Using Latent Dirichlet Allocation

Submitted to the faculty of the Electrical Engineering Department
of the University of Engineering and Technology Lahore
in partial fulfillment of the requirements for the Degree of

Bachelor of Science
in
Electrical Engineering.

Internal Examiner

External Examiner

Director
Undergraduate Studies

Department of Electrical Engineering
University of Engineering and Technology Lahore

Declaration

I declare that the work contained in this thesis is my own, except where explicitly stated otherwise. In addition this work has not been submitted to obtain another degree or professional qualification.

Signed: _____

Date: _____

Acknowledgments

We owe an ample amount of gratitude to our Instructor, Dr. Kashif Javed for the indispensable supervision they rendered us in this undertaking. Without them and the unceasing support of our faculty members and colleagues, this venture would not have been possible...

Dedicated to our instructors...

Contents

Acknowledgments	iii
List of Figures	vii
Abbreviations	viii
Abstract	ix
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	1
1.3 Organization	2
2 Literature Review	3
2.1 Latent Semantic Analysis	3
2.2 Probabilistic Latent Semantic Analysis	5
2.3 Latent Dirichlet Allocation	5
3 Software Implementation	7
3.1 Google Colaboratory	7
3.2 Dataset and Importing	7
3.3 Toolkits	8
3.3.1 Gensim	8
3.3.2 NLTK	8
3.3.3 Wordnet and Synset	9
3.3.4 pyLDAvis	9
3.4 Data Pre-processing	9
3.4.1 Stopwords	9
3.4.2 Tokenization	9
4 Results and Discussion	10
4.1 Seaborn based Text Analysis	10
4.2 Filtering of Common Words	11
4.3 Gensim based LDA Model Reults	11
4.4 pyLDAvis based Visualization of the Output	12
5 Conclusion and Future Work	15
5.1 Conclusion	15
5.2 Future Work	15

References

18

List of Figures

2.1	tf-idf score	3
2.2	Document Matrix Frmula and Hyper-Parameters	4
2.3	Dimensionalities of each matrix	4
2.4	pLSA Model	5
2.5	Joint Probability Function	5
2.6	Certain Probability Flows	6
2.7	LDA Model	6
3.1	Importing Data from Drive	7
3.2	Generating LDA Model via Gensim	8
3.3	Importing Stopwords via NLTK Library	8
3.4	Importing Wordnet	9
3.5	Code Snippet for Data Visualization via pyLDAvis	9
3.6	Tokenization	9
4.1	Text Length Density Plot	10
4.2	Text Title Density Plot	10
4.3	Count of the unique words	11
4.4	Frequency of the unique words	11
4.5	LDA Based Top 10 Recurrent Topics	11
4.6	Topics and Corresponding Weights	12
4.7	pyLDAvis Map	12
4.8	Top Topics of all articles	13
4.9	Topic 6 Selected	13
4.10	Topic 6 Line Chart	14

Abbreviations

LDA	L atent D irichlet A llocation
LSA	L atent S emantic A nalysis
pLSA	p robalistic L atent S irichlet A nalysis

Abstract

Progressively, the management specialists are utilizing topic modelling, another strategy acquired from software engineering, to uncover marvel based builds and grounded reasonable connections in literary information. By conceptualizing point displaying as the way toward delivering develops and calculated connections from printed information, we exhibit how this new technique can propel the board grant without transforming topic modelling into a black box of complex PC driven calculations. We start by contrasting highlights of point demonstrating with related strategies (content investigation, grounded estimating, and characteristic language preparing). We at that point stroll through the means of delivering with theme displaying and apply delivering to the executives articles that draw on topic modelling. Doing so empowers us to distinguish and examine how topic modelling has progressed the board hypothesis in five regions: identifying oddity and rise, creating inductive characterization frameworks, understanding on the web crowds and items, breaking down edges and social developments, and understanding social elements. In this project, the most useful technique of topic Modelling LDA was implemented.

Chapter 1

Introduction

Topic modelling is the technique of assigning any article, journal or paper a suitable title via the natural language processing.

1.1 Motivation

A lot of information are gathered ordinary. As more data opens up, it gets hard to get to what we are searching for. Thus, we need apparatuses and methods to put together, look and comprehend tremendous amounts of data.

Topic Modelling furnishes us with strategies to coordinate, comprehend and sum up enormous assortments of literary data. It helps in:

- *Finding shrouded effective examples that are available across the assortment.*
- *Commenting on reports as indicated by these subjects.*
- *Utilizing these explanations to sort out, look and sum up writings.*

So in order to cover the large number of data and articles efficiently and in a very little time, the process of topic modelling is to of great significance.

1.2 Problem Statement

Let us consider an industry working in the field of bioinformatics. topic modelling is a valuable technique (rather than the conventional methods for information decrease in bioinformatics) and improves scientists' capacity to decipher organic data. All things considered, because of the absence of theme models enhanced for explicit organic information, the examinations on point demonstrating in natural information actually have a long and testing street ahead. Lately, we have been seeing dramatic development of organic information, for example, micro-array data-sets.

The present circumstance additionally represents an extraordinary test, to be specific, how to extricate concealed information and relations from these information. As referenced above, theme models have arisen as a viable strategy for finding helpful structure

in assortments. Subsequently, a developing number of specialists are starting to incorporate subject models into different organic information, not just record assortments. In these examinations, we find that subject models go about as in excess of a grouping or bunching approach. They can show a natural article as far as covered up "themes" that can mirror the fundamental organic significance all the more thoroughly. In this way, point models were as of late demonstrated to be a useful asset for bio-informatics.

1.3 Organization

The organization of this report is as follows. Chapter 1 introduces the motivation to use topic modelling and clearly state the problem statement in bio-informatics. Chapter 2 provides a brief description of Latent Dirichlet Allocation and how it works. Chapter 3 covers the implementation and the code of the technique as well as explanation of the data-set and necessary means of implementation. In Chapter 4 , we discuss all the results of the articles in our data-set. Chapter 5 concludes our work and gives its future prospects.

Chapter 2

Literature Review

Over the course of time, different technique have been utilized for topic modelling. The theoretical explanation of them all can be seen below:

2.1 Latent Semantic Analysis

Latent Semantic Analysis, or LSA, is one of the fundamental methods in topic demonstrating. The center thought is to take a network of what we have — reports and terms — and deteriorate it into a different archive topic matrices and a subject-term matrix. The initial step is creating our report-topic matrix. Given m reports and n words in our articles, we can build a $m \times n$ matrix A in which each line speaks to a record and every segment speaks to a word. In the least complex rendition of LSA, every section can basically be a crude check of the occasions the j -th word showed up in the I -th report. Practically speaking, be that as it may, crude checks don't function admirably on the grounds that they don't represent the centrality of each word in the archive. For instance, "atomic" most likely educates us more about the topic(s) of a given archive than "test." Subsequently, LSA models normally supplant crude includes in the record term network with a tf-idf score. Tf-idf, or term frequency inverse document frequency, appoints a load for term j in report I as follows:

The diagram shows the formula $w_{i,j} = tf_{i,j} \times \log \frac{N}{df_j}$ with four color-coded annotations and arrows:

- A green arrow points from "# occurrences of term in document" to $tf_{i,j}$.
- A red arrow points from "tf-idf score" to the entire formula.
- A blue arrow points from "# total documents" to N .
- A purple arrow points from "# documents containing word" to df_j .

$$w_{i,j} = tf_{i,j} \times \log \frac{N}{df_j}$$

FIGURE 2.1: tf-idf score

Intuitively, a term has an immense weight when it happens frequently across the report anyway seldom across the corpus. "Fabricate" may appear as often as possible in a report, yet since it's apparent truly typical in the rest of the corpus, it won't have a high tf-idf score. In any case, if "improvement" appears every now and again in a report, since it is more unprecedented in the rest of the corpus, it will have a higher tf-idf score. At the point when we have our report term cross section A, we can start thinking about our dormant focuses. Stop and think for a moment: most likely, A is especially deficient, noisy, and tedious across its various estimations. In this manner, to find the couple of dormant subjects that get the associations among the words and reports, we need to perform dimensionality decline on A.

This dimensionality abatement can be performed using abbreviated SVD. SVD, or singular worth disintegration, is a strategy in direct factor based mathematical that factorizes any organization M into the aftereffect of 3 separate cross sections: $M=U*S*V$, where S is a cockeyed system of the specific assessments of M. In a general sense, abbreviated SVD diminishes dimensionality by picking simply the t greatest single characteristics, and simply keeping the primary t sections of U and V. For the present circumstance, t is a hyperparameter we can pick and adjust to reflect the amount of focuses we need to find.

$$A \approx U_t S_t V_t^T$$

FIGURE 2.2: Document Matrix Formula and Hyper-Parameters

Intuitively, think of this as only keeping the t most significant dimensions in our transformed space.

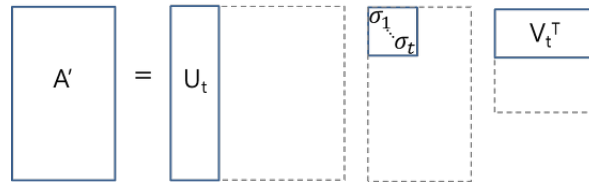


FIGURE 2.3: Dimensionalities of each matrix

For this situation, U belongs to $\mathbb{R}^{(m \times t)}$ arises as our matrix, and V belongs to $\mathbb{R}^{(n \times t)}$ ($n \times t$) turns into our term-topic matrix. In both U and V, the sections compare to one of our t points. In U, columns speak to archive vectors communicated as far as points; in V, rows speak to term vectors communicated regarding themes.

2.2 Probabilistic Latent Semantic Analysis

pLSA, or Probabilistic Latent Semantic Analysis, uses a probabilistic method instead of SVD to deal with the issue. The middle idea is to find a probabilistic model with lethargic subjects that can make the data we find in our record term grid. In particular, we need a model $P(D,W)$ with the ultimate objective that for any report d and word w , $P(d,w)$ analyzes to that segment in the record term structure.

Audit the key assumption of subject models: each chronicle involves a mix of topics, and each point includes a variety of words. pLSA adds a probabilistic go to these notions:

- given a record d , subject z is accessible in that file with probability $P(z|d)$
- given a subject z , word w is drawn from z with probability $P(w|z)$

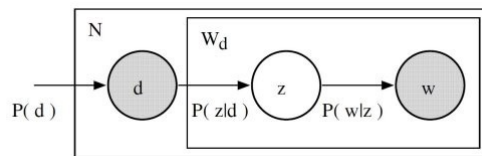


FIGURE 2.4: pLSA Model

Formally, the joint probability of seeing a given file and word together is:

$$P(D,W) = P(D) \sum_z P(Z|D)P(W|Z)$$

FIGURE 2.5: Joint Probability Function

Despite the way that it looks exceptionally changed and moves toward the issue in an inside and out various way, pLSA genuinely adds a probabilistic treatment of focuses and words on top of LSA. It is a verifiably more versatile model, yet has several issues. Explicitly considering the way that we have no limits to show $P(D)$, we don't have the foggiest thought how to give out probabilities to new chronicles and the amount of limits for pLSA grows straightly with the amount of reports we have, so it is slanted to over-fitting.

2.3 Latent Dirichlet Allocation

LDA is a Bayesian transformation of pLSA. In particular, it uses dirichlet priors for the file subject and word-point scatterings, fitting better hypothesis.

Consider the amazingly relevant delineation of taking a gander at probability movements of point mixes. Assume the corpus we are looking at has records from 3 inside and out various parts of information. If we need to show this, the sort of scattering we need will be one that seriously stacks one express topic, and doesn't give a ton of weight to the rest in any way shape or form. If we have 3 subjects, by then some specific probability

flows we'd likely notice are:

- **Mixture X:** 90% topic A, 5% topic B, 5% topic C
- **Mixture Y:** 5% topic A, 90% topic B, 5% topic C
- **Mixture Z:** 5% topic A, 5% topic B, 90% topic C

FIGURE 2.6: Certain Probability Flows

If we draw a discretionary probability scattering from this dirichlet dissemination, characterized by colossal burdens on a singular subject, we would presumably get a movement that insistently takes after either mix X, mix Y, or mix Z. It would be unrealistic for us to test an allotment that is 33% subject A, 33% topic B, and 33% point C. In pLSA, we test a report, by then a point reliant on that record, by then a word subject to that point. Here is the model for LDA:

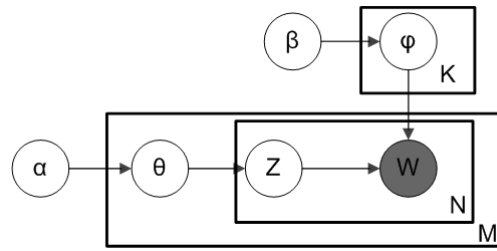


FIGURE 2.7: LDA Model

From a dirichlet transport $\text{Dir}(\alpha)$, we draw an unpredictable model addressing the point appointment, or subject mix, of a particular record. This point movement is . From , we select a particular subject Z reliant on the allocation.

Next, from another dirichlet transport $\text{Dir}(\beta)$, we select an unpredictable model addressing the word dissemination of the subject Z . This word flow is denoted by symbol ϕ . From this, we pick the word w .

LDA usually works in a manner that is superior to pLSA considering the way that it can summarize to new records with no issue. In pLSA, the file probability is a fixed point in the dataset. If we haven't seen a report, we don't have that data point. In LDA, the dataset fills in as planning data for the dirichlet scattering of report point scatterings. In case we haven't seen a record, we can without a doubt test from the dirichlet transport and push ahead starting there.

Chapter 3

Software Implementation

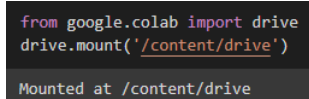
In this chapter, we shall discuss the necessary toolkit and the software implementation of our project which we have done.

3.1 Google Colaboratory

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. It is the environment used to write the code.

3.2 Dataset and Importing

The dataset used for the implementation of topic modelling was an articles dataset in a csv file where all the articles were provided from medium. Medium is one of the most famous tools for spreading knowledge about almost any field. It is widely used to publish articles on ML, AI, and data science. This dataset is the collection of about 350 articles in such fields. The dataset contains articles, their title, number of claps it has received, their links and their reading time. This dataset was scraped from Medium. I created a Python script to scrap all the required articles using just their tags from Medium. The dataset was then saved on google drive and imported to collab. The importing of the dataset can be seen as follows:



```
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
```

FIGURE 3.1: Importing Data from Drive

3.3 Toolkits

In this section, let us go through some of the important toolkits used in implementation of the task. Let us go through them as follows:

3.3.1 Gensim

Gensim means "Generate Similar" is a mainstream open source natural language processing (NLP) library utilized for solo point demonstrating. It utilizes top scholarly models and current factual AI to perform different complex assignments, for example:

- Building record or word vectors
- Corpora
- Performing theme ID
- Performing record examination (recovering semantically comparative archives)
- Investigating plain-text records for semantic structure

Aside from playing out the above complex assignments, Gensim, actualized in Python and Cython, is intended to deal with huge content assortments utilizing information gushing just as steady online calculations. This makes it not quite the same as those AI programming bundles that target just in-memory preparing.

```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                             id2word=dictionary,
                                             num_topics=10,
                                             random_state=100,
                                             update_every=1,
                                             chunksize=500,
                                             passes=20,
                                             alpha='auto',
                                             per_word_topics=True)
```

FIGURE 3.2: Generating LDA Model via Gensim

3.3.2 NLTK

NLTK is one of the main stages for working with human language interactions and in Python, the module NLTK is utilized for characteristic language preparing. NLTK is in a real sense an abbreviation for Natural Language Toolkit.

```
nltk.download('stopwords')
```

FIGURE 3.3: Importing Stopwords via NLTK Library

3.3.3 Wordnet and Synset

WordNet is the lexical information base for example word reference for the English language, explicitly intended for characteristic language preparing.

Synset is an exceptional sort of a straightforward interface that is available in NLTK to look into words in WordNet. Synset occasions are the groupings of interchangeable words that express a similar idea. A portion of the words have just a single Synset and some have a few.

```
nltk.download('wordnet')
docs = [words(x) for x in data['text']]
```

FIGURE 3.4: Importing Wordnet

3.3.4 pyLDAvis

pyLDAvis is intended to assist clients with interpreting the subjects in a theme model that has been fit to a corpus of text information. The bundle separates data from a fitted LDA subject model to advise an intelligent electronic perception.

```
pyLDAvis.enable_notebook()
warnings.filterwarnings("ignore", category=DeprecationWarning)
```

FIGURE 3.5: Code Snippet for Data Visualization via pyLDAvis

3.4 Data Pre-processing

For the pre processing of the data we have used, we use tokens and stopwords.

3.4.1 Stopwords

Stopwords are the English words which doesn't add a lot of importance to a sentence. They can securely be overlooked without relinquishing the importance of the sentence. For instance, the words like the, he, have and so forth Such words are now caught this in corpus named corpus. We initially download it to our python climate.

3.4.2 Tokenization

Tokenization is basically parting an expression, sentence, passage, or a whole book record into more modest units, for example, singular words or terms. Every one of these more modest units are called tokens.

Natural Language Processing
['Natural', 'Language', 'Processing']

FIGURE 3.6: Tokenization

Chapter 4

Results and Discussion

In this chapter, we shall discuss the important points of the implementation and their corresponding results.

4.1 Seaborn based Text Analysis

Firstly, once the data-set has been imported into the notebook, the analysis of the length of the text in each of the articles is observed. It is done to see the density of the text in the corresponding sections.

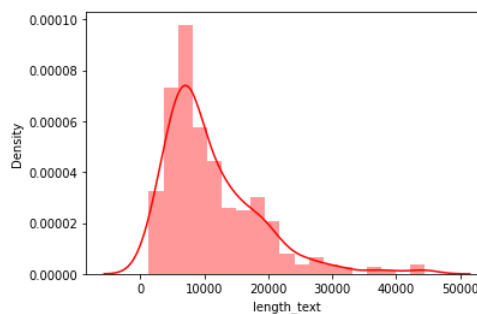


FIGURE 4.1: Text Length Density Plot

Similarly, the analysis is also performed on the title of each of the articles used in the dataset and its corresponding density plot can be seen as follows:

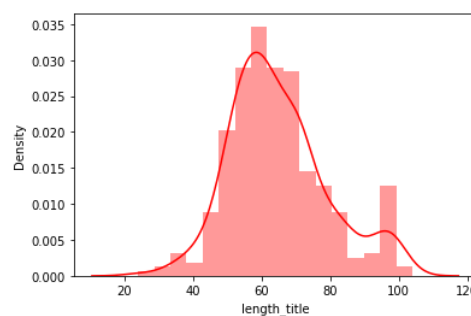


FIGURE 4.2: Text Title Density Plot

4.2 Filtering of Common Words

Next up, the document has many words like the, and, so, that and many more. These words are referred to as stopwords and are not essential in the analysis regarding the topic of the article so all these words and rare words are filtered out and we can see these as follows: Now the unique words as identified can also be viewed along with the

```
# Create a dictionary representation of the documents.
dictionary = Dictionary(docs)
print('Number of unique words in initial documents:', len(dictionary))

# Filter out words that occur less than 10 documents, or more than 20% of the documents.
dictionary.filter_extremes(no_below=10, no_above=0.2)
print('Number of unique words after removing rare and common words:', len(dictionary))

Number of unique words in initial documents: 18975
Number of unique words after removing rare and common words: 2720
```

FIGURE 4.3: Count of the unique words

number of times that particular word repeats itself and it can also be seen as follows:

```
for i in range(len(bow_doc_300)):
    print("Word {} (\{}\)\ appears {} time.".format(bow_doc_300[i][0],
                                                    dictionary[bow_doc_300[i][0]],
                                                    bow_doc_300[i][1]))

Word 47 ("close") appears 1 time.
Word 51 ("command") appears 1 time.
Word 65 ("cool") appears 1 time.
Word 121 ("gate") appears 1 time.
Word 189 ("multi") appears 1 time.
Word 213 ("predicted") appears 1 time.
Word 247 ("scene") appears 1 time.
Word 281 ("terminal") appears 1 time.
```

FIGURE 4.4: Frequency of the unique words

4.3 Gensim based LDA Model Results

Gensim library is used to build an LDA model and then the LDA model provides results on all the data it has observed as follows: The algorithm does the same for all

```
get_lda_topics(lda_model, 10)
```

	Topic # 01	Topic # 02	Topic # 03	Topic # 04	Topic # 05	Topic # 06	Topic # 07	Topic # 08	Topic # 09	Topic # 10
0	neuron	star	cpu	de	cluster	bot	car	table	pixel	cnn
1	activation	review	house	member	agent	conversation	music	reward	woman	sheet
2	matrix	rating	gtx	sound	policy	behavior	graph	startup	men	box
3	zero	average	sentence	title	host	social	batch	recognize	convolution	region
4	player	weighted	gpu	speech	response	simulation	vehicle	array	kernel	pixel
5	policy	university	sequence	et	lstm	apps	track	tutorial	facial	bounding

FIGURE 4.5: LDA Based Top 10 Recurrent Topics

the articles but as the output will be too big, we have only shown a few here.

Now, an important aspect that this above code misses is why a particular word is above the rest.

For this aspect the code below is used: The above code gives the top suggested topics

```
lda_model.save('model10.gensim')
topics = lda_model.print_topics(num_words=6)
for topic in topics:
    print(topic)

(0, '0.019*neuron' + 0.010*activation' + 0.009*matrix' + 0.008*zero' + 0.008*player' + 0.008*policy')
(1, '0.017*star' + 0.015*review' + 0.012*rating' + 0.011*average' + 0.010*weighted' + 0.010*university')
(2, '0.010*cpu' + 0.010*house' + 0.009*gtx' + 0.009*sentence' + 0.009*gpu' + 0.008*sequence')
(3, '0.034*de' + 0.014*member' + 0.012*sound' + 0.011*title' + 0.009*speech' + 0.009*et')
(4, '0.014*cluster' + 0.012*agent' + 0.008*policy' + 0.007*host' + 0.007*response' + 0.007*lstn')
(5, '0.011*bot' + 0.006*conversation' + 0.005*behavior' + 0.005*social' + 0.005*simulation' + 0.004*apps')
(6, '0.011*car' + 0.010*music' + 0.008*graph' + 0.008*batch' + 0.007*vehicle' + 0.006*track')
(7, '0.023*table' + 0.009*reward' + 0.008*startup' + 0.007*recognize' + 0.007*array' + 0.006*tutorial')
(8, '0.020*pixel' + 0.018*woman' + 0.016*men' + 0.015*convolution' + 0.015*kernel' + 0.011*facial')
(9, '0.095*cnn' + 0.035*sheet' + 0.035*box' + 0.034*region' + 0.021*pixel' + 0.020*bounding')
```

FIGURE 4.6: Topics and Corresponding Weights

for each article along with its particular weight and the word which is more suitable to be the topic has the highest weight as can be seen above.

4.4 pyLDAvis based Visualization of the Output

As described in an earlier section, pyLDAvis is a library which provides visualization of the LDA model that has been devised. It provides a map where one can see all the topics with relevance to the overlapping between the words all the provided articles in the dataset show.

It can be seen as follows:

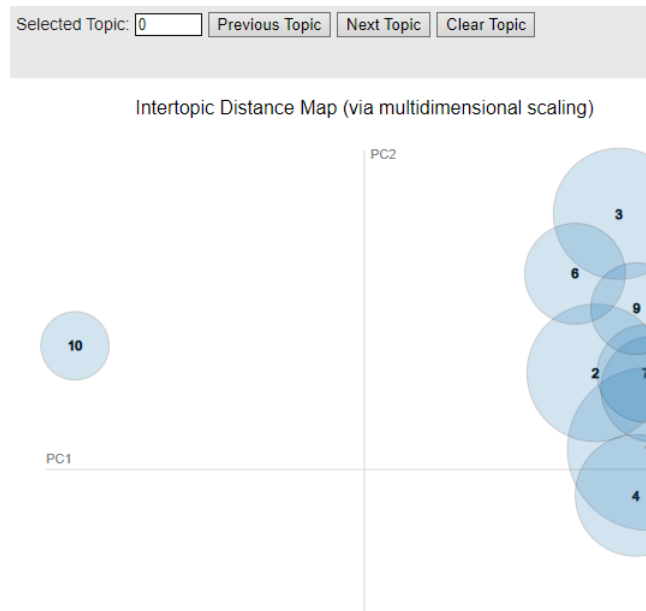


FIGURE 4.7: pyLDAvis Map

Here you can see all the topics in a map represented by their corresponding circles and on basis of the common text between the articles each of the article is given its relevant overlapping.

Below can be seen the top words on basic of which our algorithm builds or models the topic of a particular article.

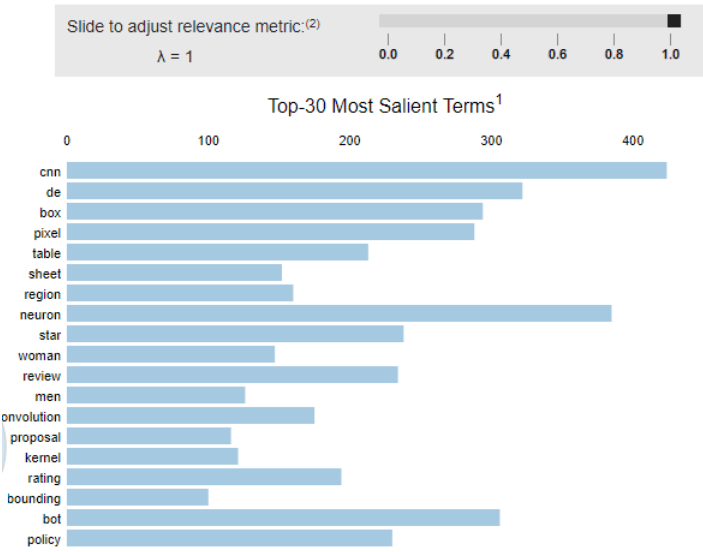


FIGURE 4.8: Top Topics of all articles

Similarly if we choose a particular topic i.e topic 6 as in the below figure, then the algorithm shows the selected topic highlighted as follows:

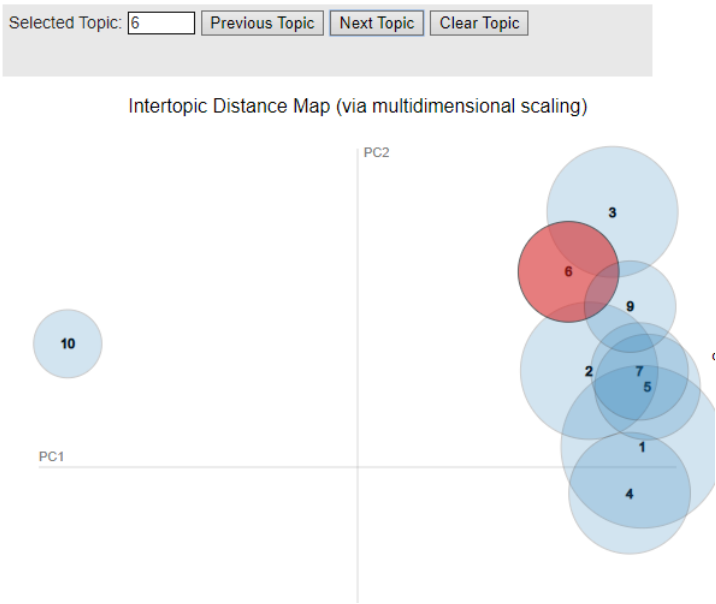


FIGURE 4.9: Topic 6 Selected

Also, if we look at the line chart for the top topics we see that it now shows all the topics of article number 6 and their respective weightage in the article.

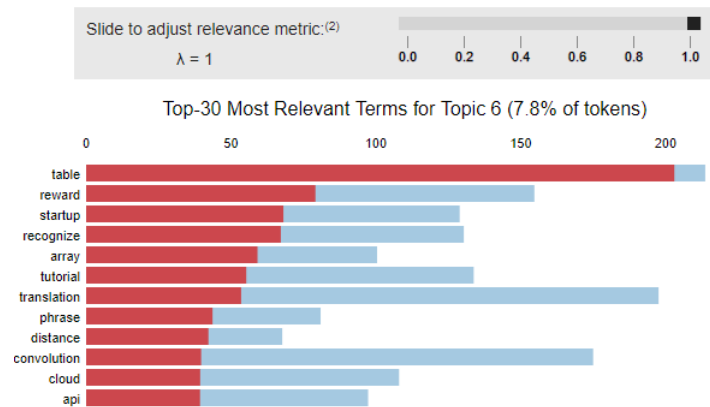


FIGURE 4.10: Topic 6 Line Chart

This is how latest python libraries like gensim and NLTK are working in topic modelling.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Firstly, in this implementation we looked at the techniques which re mainly used in topic modelling. A summary of these is also shown below. After the understanding of the methods, we implemented the LDA algorithm on the dataset we devised via articles and journals on data science and performed analysis on the dataset via gensim and NLTK libraries. At the end, we concluded our work by going through each of the results and then the visualization of the entire data via pyLDAvis library through which we saw a map based representation of our data.

Name of The Methods	Characteristics
Latent Semantic Analysis (LSA)	<ul style="list-style-type: none">* LSA can get from the topic if there are any synonym words.* Not robust statistical background.
Probabilistic Latent Semantic Analysis (PLSA)	<ul style="list-style-type: none">* It can generate each word from a single topic; even though various words in one document may be generated from different topics.* PLSA handles polysemy.
Latent Dirichelet Allocation (LDA)	<ul style="list-style-type: none">* Need to manually remove stop-words.* It is found that the LDA cannot make the representation of relationships among topics.

5.2 Future Work

Today, topic modelling is still generally reliant on two figurings, LSA and LDA. These models are mathematically steady and gainful for subject showing purposes. Nevertheless, they make them glare deformity: they are sans setting. The frequencies of words and articulations scattered all through a corpus is the way these computations work, anyway frequencies don't give meaning.

For example, since topic modelling is out and out independent learning, we can't reason assessment from a point model. We would need to run that examination freely, using full scale controlled learning methods. Today, this level of examination is truth be told possible anyway excessively far for the ordinary business customer; just purchasers and

customers of excellent quality language exhibiting programming approach it. Topic Modelling will create to a significant learning model that sees instances of setting and assessment, similarly as melding the current mathematical systems. It may, dependent upon the improvement of significant learning developments, even join PC vision, to see the spatial thought of text and its employment in inclination and tone.

References

- [1] Medium. 2020. Topic Modelling In Python With NLTK And Gensim. [online] Available at: <https://towardsdatascience.com/topic-modelling-in-python-with-nltk-and-gensim-4ef03213cd21>.
- [2] S. Li, “Topic Modeling and Latent Dirichlet Allocation (LDA) in Python,” Medium, 01-Jun-2018. [Online]. Available: <https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>.
- [3] R. Chawla, “Topic Modeling with LDA and NMF on the ABC news headlines dataset,” ML 2 Vec, 30-Jul-2017. [Online]. Available: <https://medium.com/ml2vec/topic-modeling-is-an-unsupervised-learning-approach-to-clustering-documents-to-discover-topics-fdfbf30e27df>.
- [4] C. B. Asmussen and C. Møller, “Smart literature review: a practical topic modelling approach to exploratory literature review,” J. Big Data, vol. 6, no. 1, 2019.
- [5] P. Kherwa and P. Bansal, “Topic modeling: A comprehensive review,” ICST Trans. Scalable Inf. Syst., vol. 0, no. 0, p. 159623, 2018.
- [6] M. Reisenbichler and T. Reutterer, “Topic modeling in marketing: recent advances and research opportunities,” J. Bus. Econ., vol. 89, no. 3, pp. 327–356, 2019.
- [7] Hsankesara, “Medium Articles.” .