**Facebook Sentiment Analysis using Topic Modelling**

Bilal Abid, 11639888, [abidb@uni.coventry.ac.uk](mailto:abidb@uni.coventry.ac.uk)

Chukwuemeka Obinna Emodi, 10067480, [emodic@uni.coventry.ac.uk](mailto:emodic@uni.coventry.ac.uk)

Faculty of Engineering, Environment and Computing, Coventry University

MSc Data Science and Computational Intelligence (EECT044) Stage 1

7135CEM – Modelling and Optimization Under Uncertainty

Coventry, United Kingdom

***Abstract – This paper investigates the study about two different topic modelling algorithms, Latent Dirichlet Allocation and Latent Semantic Analysis, that are being used in machine learning and natural language processing for data mining and sentiment analysis applications. This focus of this study is to implement these two models on Facebook data to analyze the sentiments of its users. And finally, to perform a comparative analysis between the two models, to see which model gave optimized results.***

***Keywords – Topic modelling; Latent Dirichlet Allocation – LDA; Latent Semantic Analysis – LSA; machine learning; natural language processing – NLP; Data mining; sentiment analysis; Facebook.***

# **INTRODUCTION**

Over the past decade, the amount of data we had produced is beyond comprehension. Around 4 petabyte of data per day (a million gigabytes) is being generated by Facebook alone in form of post, comments, photos, and videos [1]. But this paper will be focused on analyzing textual data only. Sentiment analysis is being used by Facebook and other social media platforms to monitor the attitude, feelings and reviews people have towards different brands and companies. But is it positive or negative ? To figure that out you need to analyze those specific conversations. Facebook does that by collecting, carefully analyzing, and extracting useful information in these posts people share about different brands on its platform. Furthermore, opinions shared on social media platform are provided as a feedback for the business model as it helps in understanding which area of business excels and which needs further improvements.

Topic modelling is an unsupervised machine learning technique used for text mining and natural language processing (nlp); it is a statistical model that extracts latent semantic structures from a corpus of documents. Large amounts of textual data can be organized using topic modelling and provides insight for understanding large amount of unstructured text data. The two most important topic modelling techniques used for text mining and sentiment analysis are Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis which we will be using in this experiment for Facebook sentiment analysis.

# **DATASET**

The dataset used in this paper is fetched from the Kaggle repository and is available online. The name of the dataset file is “**fb\_sentiment**”, it has 2 columns (“**FBPost**”, and “**Label**”) and 1000 rows. Each row represents whether the subsequent post has positive (**P**), negative (**N**) or some other (**O**) sentiments. A direct link to this dataset is provided in the appendix section. Figure-1 shows the following dataset.

A picture containing Word

Description automatically generated

Figure 1: Facebook Sentiment Dataset

It can be observed from the above figure that all posts in the dataset are curated around a specific Amazon product “Kindle”. Kindle is an e-book reader developed and sold by Amazon exclusively, and allows its users to browse, buy, download, and read e-books, magazines, dictionaries, and other digital media either on Amazon Kindle e-reader specialized tablets or on smart phones and computers.

# **METHODS**

The two methods we will be using for topic modelling in this paper are Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA).

1. Latent Dirichlet Allocation

LDA is one of the most promising topic modelling technique used for extracting topics and keywords from a corpus of text documents. And topics or themes are usually hidden within this data, which needs to be surfaced i.e., Latent. Whereas Dirichlet Allocation originates from Dirichlet distribution and belongs to a family of stochastic probability distribution and processes.

LDA can be defined as the distribution of distributions whereas, in context of topic modelling it generates a set of frequently repeating keywords which are related to each other. LDA model provides us the probability distribution of words P(words | topics) from which topics and themes are derived, and these documents are then further classified into distribution of topics P(topics | documents). These probabilities are first calculated randomly, then an iteration process is initialized that recomputes these probability values according to their topic until a certain convergence point is met.

Diagram

Description automatically generated with low confidence

Figure 2: LDA document per topic and topic per words distributions [2]

Diagram

Description automatically generated

Figure 3: LDA Model [3]

Figure 3 shows the total number of documents (M), frequency of words in document (N), word in document is represented by (w), (z) is the latent/hidden assigned topic to word, theta (Ɵ) gives the topic distribution. Whereas LDA has 2 hyperparameters α and β used for controlling or tuning the similarity of documents. An increasing value of alpha generates more topics assigned per document, and a greater value of beta generates topic using more keywords and vice-versa.

1. Latent Semantic Analysis

Latent semantic analysis uses a bag of words (BoW) model to create a document-term matrix for topic modelling and it ignore the semantics and syntactics as the model is considered a bag of words. It calculates the document and word frequency by using TF-IDF metrics, where (TF) is the term frequency and (IDF) is the inverse document frequency across a set of whole documents. TF computes the word frequency for each document and is only limited to each singular document instead of the whole corpus. Therefore, we use the IDF metric to calculate how significant each term is across the whole corpus of documents.

In LSA extraction of latent topics is done by applying Singular Value Decomposition (SVD) on the document-term matrix for decomposing this matrix into a product of 3 matrices as shown in figure 4, where each term is a row vector and documents represents the column vectors [4].

Table, timeline

Description automatically generated

Figure 4: SVD Matrix decomposition [4]

# **EXPERIMENT**

As mentioned earlier in this experiment we will be using both LDA and LSA for our topic modelling feature extraction to analyze which model perform better in term of sentiment analysis. Here all the programming were done in Python 3 on Google Colab, and a direct link to complete code can be found in the appendix section.

Steps involved in this experiment are,

1. Loading python libraries
2. Exploratory data analysis (EDA)
3. Data Pre-processing
4. LDA and LSA modelling and analysis
5. Loading Python Libraries

In this experiment we used pandas, numpy packages for loading and manipulating dataset structures, whereas genism was the main library used for topic modelling. Whereas libraries used for data visualization were matplotlib, scipy, t-SNE, pyLDAvis, textblob and wordcloud.

1. Exploratory Data Analysis

Before putting the data through our pre-processing system, first we need to visualize this data from different perspectives to make some observations and gather insights for our given dataset.

Figure 5 shows 15 top-most keywords that are frequently occurring in our given dataset, with “kindle”, “love”, “book” and “read” being the most prominent words. And it clearly indicates that the context of our given dataset is related to Amazon Kindle.

Whereas figure 6 gives us the distribution for part-of-speech tags (POSTags) across the whole corpus of documents and shows what tags each of these words are carrying, with nouns (NN), pronoun (PRP) being the top tags used in this dataset.

Chart, bar chart, histogram

Description automatically generated

Figure 5: Top-most frequently used words

Chart, bar chart, histogram

Description automatically generated

Figure 6: POS Tagging

1. Data Pre-Processing

Figure 7 shows samples from our dataset, and it can be observed that our dataset has a lot of data inconsistencies like website links and URL’s, punctuations, uneven white spaces, user tags, unnecessary stopwords, and spelling mistakes.

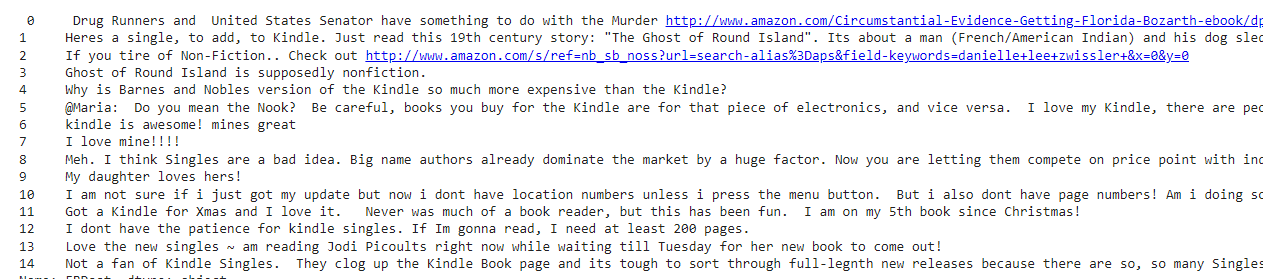


Figure 7: Raw data before pre-processing

We can get rid of all these inconsistencies within our data with the help of some pre-processing libraries (re, sample\_preprocess, nltk, spacy). First we removed these inconsistencies, performed word tokenization, removed stopwords, build bigram and trigrams words, and then performed lemmatization by using those libraries.

The wordcloud shown in figure 8 shows the pre-processed data keywords.

Text

Description automatically generated

Figure 8: Pre-processed data wordcloud

1. LDA and LSA Modelling and Analysis
2. Latent Dirichlet Allocation

First, we began by creating the corpus dictionary, then we used genism library for building our lda model where we have selected 8 number of topics in our model, whereas the values of beta and eta are set to auto. These two hyperparameter values are used for controlling the document similarity. Top keywords for all 8 topics are represented in figure 9.

Table

Description automatically generated

Figure 9: Topics keywords

We know that a document is a combination of multiple topics, but out of all those topics only one might be the dominant one for each document as shown in figure 10. And we noticed that although topic 7 has a 50 percent contribution for document 0, but still none of those topic keywords were relevant to the corresponding document. And the same was observed for all the other documents. However, after manipulating and playing around with this data frame it was observed that in figure 10 the data frame was organized and indexed according to document numbers and after indexing the data frame with respect to “Perc\_Contribution” in a descending order thing started to make sense in our data as shown in figure 11.

For example, if we look at the seventh row the context of that document is talking about,

I bought Kindles for 2 sisters who love to read; **keywords: family, buy**

And getting books are easier than going to bookstores; and cheaper; **keywords: download**

Graphical user interface, text, application

Description automatically generated with medium confidence

Figure 10: Per Topic Percentage Contribution

A picture containing diagram

Description automatically generated

Figure 11: Per Topic Percentage Contribution after proper indexing

Figure 12 represents the distribution of documents by word counts across the whole corpus, and in figure 13 we created a word cloud for representing each topic keywords.

Chart, histogram

Description automatically generated

Figure : Distribution of Documents word counts

Text

Description automatically generated with low confidence

Figure 13: Per Topic Word Cloud

Here we have used two different kind of graphical representations that shows the importance of top 10 then top 3 topics keywords by their word counts and how much weightage each keyword has in their respective topics and across the whole corpus, as shown in figures 14 and 15. And (work, kindle and send) keywords from topic 4 are the most dominant keywords with more weightage as compared to others.

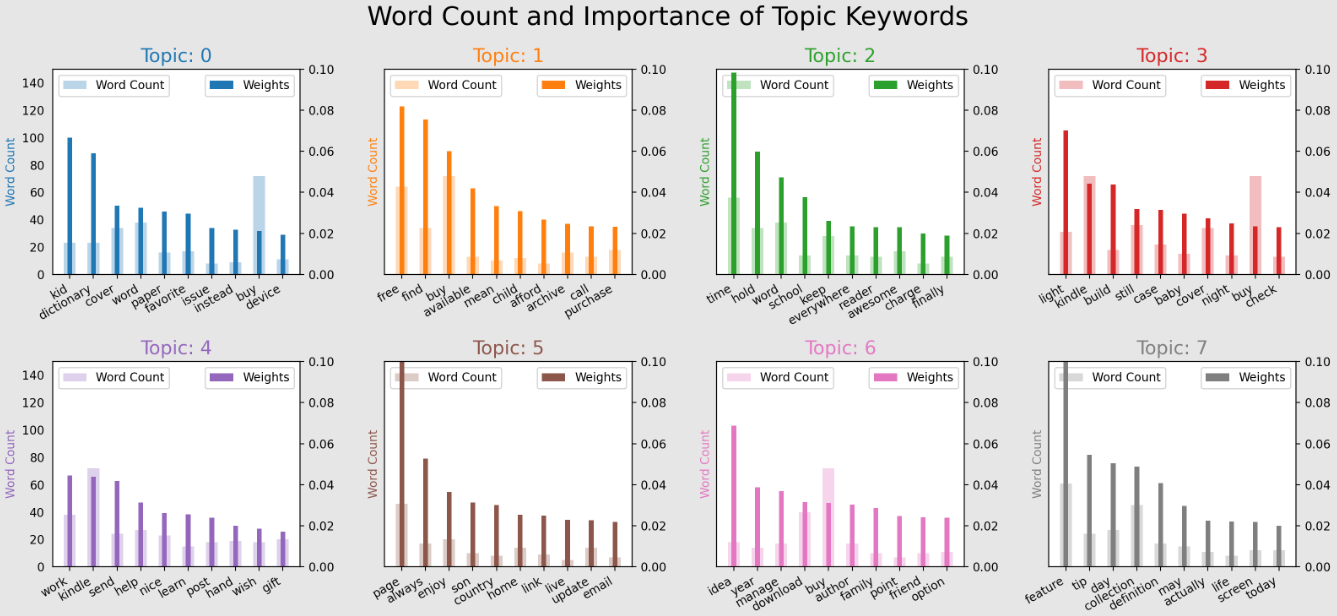


Figure 14: Word Count and Importance of Topic Keywords

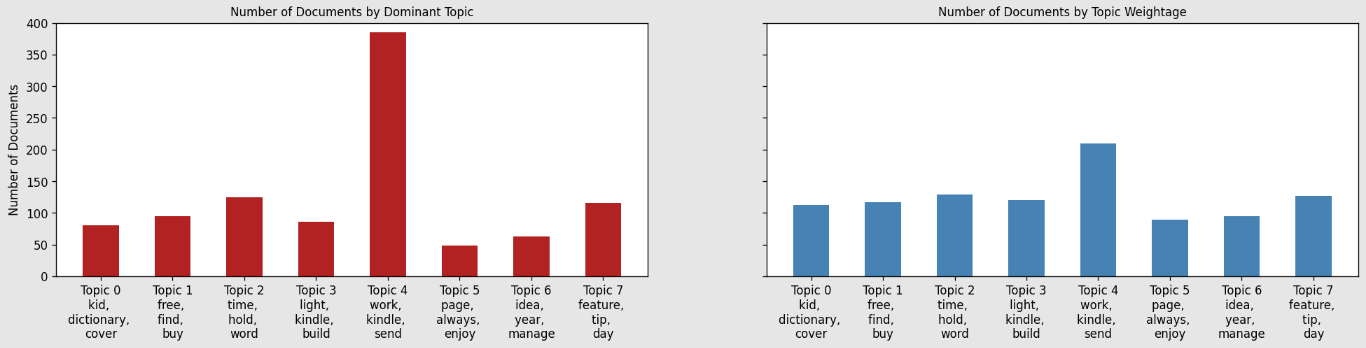


Figure 15: Number of Documents by Dominant Topics and Weightage

Now we will be using t-distributed Stochastic Neighbor Embedding (t-SNE) as a dimensionality reduction algorithm for visualizing our highly dense multi-dimensional data into two dimensions. And it can be analyzed from figure 16 that all our topics are very distinctly separated from each other. Finally, were going to use another technique called “pyLDAvis” for mapping these clusters as shown in figure 17. It can be seen that topics 1,2,3,5 and 8 are separated from each other, while topics 4,6 and 7 are slightly overlapping each other because some of keywords (buy, husband, etc.) are jointly shared by these topic clusters.

Chart, scatter chart

Description automatically generated

Figure 16: t-SNE Clustering of 8 LDA Topics

Chart, bubble chart

Description automatically generated

Figure 17: LDA Intertopic Mapping using pyLDAvis

1. Latent Semantic Analysis

For LSA we performed the same pre-processing task that we did for LDA, after pre-processing we created two functions for extracting top keywords in our topics to have a better understanding of our topics. The top keywords for each topic are shown in below figure.

A screenshot of a computer

Description automatically generated with low confidence

Figure 18: LSA top 10 keywords

Chart, bar chart

Description automatically generated

Figure 19: LSA top 3 keywords

Here again were going to use t-SNE clustering technique to check how well the LSA model has performed and to visualize our topic clusters. We can analyze from figure 20 that the LSA model has not performed well for our given dataset and had failed to segregate our topics.

Chart, scatter chart

Description automatically generated

Figure 20: t-SNE Clustering of LSA Model

# **SOCIAL ETHICAL, AND LEGAL ISSUES TO CONSIDER**

Since the dataset we are using was originally extracted from Facebook and consists of different user posts, so before using this dataset we had to make sure that the privacy of user’s identities was preserved.

# **CONCLUSION**

To conclude, this paper investigates the Latent Dirichlet Allocation and Latent Semantic Analysis topic modelling techniques for analyzing user sentiments from Facebook posts. In contrast to LSA, LDA model had performed really well in separating all the topics and had extracted keywords that are relevant to each topic. Here it is important to mention the reason why the LDA model was able to outperform LSA was because of its assumption on the distribution of topics in a document along with the distribution of words. And the only issue we faced with LDA was that we had to spend a lot of time on data pre-processing, because the dataset we had used in this experiment were post extracted from Facebook and it had a lot of spelling mistakes and other discrepancies which we had to remove manually.

# **REFERENCES**

1. <https://kinsta.com/blog/facebook-statistics/>
2. <https://iq.opengenus.org/topic-modelling-techniques/>
3. <https://www.researchgate.net/figure/Graphical-model-of-latent-Dirichlet-allocation-LDA_fig1_326140642>
4. <https://www.datacamp.com/community/tutorials/discovering-hidden-topics-python>
5. <https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/#1introduction>

# **APPENDIX**

Dataset: <https://www.kaggle.com/mortena/facebook-comments-sentiment-analysis/data>

Complete python code link: <https://colab.research.google.com/drive/1mzaHdOX3KiWysbCRkK06P75JwE7wbMc4?usp=sharing>