# TEXT MINING OF TWITTER DATA: TOPIC MODELLING

**Njoku, Uchechukwu Fortune**

**(40572)**

**A Thesis submitted to the Faculty of Computer Science at the African University of Science and Technology**

**In Partial Fulfilment of the Requirements for the degree of Master of Science in the Computer Science Department**

**June 2019**

## African University of Science and Technology [AUST]

*Knowledge is Freedom*

# APPROVAL BY

**Supervisor**

Surname: Prasad

First name: Rajesh

Signature

**The Head of Department**

Surname: DAVID

First name: Amos

Signature:

# ABSTRACT

Access to the Internet is becoming more affordable especially in Africa and with this the number of active social media users is also on the rise. Twitter is a social media platform on which users post and interact with messages known as "tweets". These tweets are usually short with a limit of 280 characters. With over 100 million Internet users and 6 million active monthly users in Nigeria, lots of data is generated through this medium daily. This thesis aims to gain insights from the ever-growing Nigerian data generated from twitter using Topic modelling. We use Latent Dirichlet Allocation (LDA) on Nigerian heath tweets from verified accounts covering time period of 2015 – 2019 to derive top health topics in Nigeria. We detected the outbreaks of Ebola, Lassa fever and meningitis within this time frame. We also detected reoccurring topics of child immunization/vaccination. Twitter data contains useful information that can give insights to individuals, organizations and the government hence it should be further explored and utilized.

Keywords: Twitter, Text mining, Health, Topic modelling, Latent Dirichlet Allocation.

# DEDICATION

This work is dedicated to every young researcher, my family, friends and colleagues.

# ACKNOWLEDGEMENT

I acknowledge God for life and this opportunity. I thank my supervisor for his guidance through this work, my family for their support and my friends and colleague for always being there. I also acknowledge the African Development Bank (AfDB) for their full sponsorship throughout my master's program.

# Table of Contents

List of Figures and Tables ........ viii

1 INTRODUCTION ........ 1

1.1 Introduction ........ 1

1.2 Data Mining ........ 1

1.2.1 Data collection ........ 2

1.2.2 Feature extraction and data cleaning ........ 2

1.2.3 Analytical processing and algorithms ........ 2

1.3 Text Mining ........ 3

1.4 Topic Modelling ........ 3

1.5 Applications of Topic Modelling ........ 3

1.6 Problem Statement ........ 4

1.7 Aim and Objectives ........ 4

1.8 Methodology ........ 5

2 LITERATURE REVIEW ........ 6

2.1 Introduction ........ 6

2.2 Basic Terminologies ........ 6

2.2.1 Term/Token ........ 6

2.2.2 Document ........ 6

2.2.3 Corpus ........ 6

2.2.4 Bag of words ........ 7

2.2.5 Term frequency (TF) ........ 7

2.2.6 Inverse Document Frequency (IDF) ........ 7

2.2.7 Term Frequency–Inverse Document Frequency (TF-IDF) ........ 8

2.2.8 Document- Term matrix ........ 8

2.2.9 Document-Topic matrix ........ 8

2.2.10 Topic-Word matrix ........ 8

2.3 Topic modelling ........ 8

2.3.1 Latent Semantic Analysis (LSA) ........ 9

2.3.2 Probabilistic Latent Semantic Analysis (LSI) ........ 9

2.3.3 Latent Dirichlet Allocation (LDA) ........ 10

2.3.4 Lda2Vec ........ 10

2.4 Review of Literatures ........ 10

2.5 Conclusion ........ 19

3 METHODOLOGY ........ 20

3.1 Introduction ........ 20

3.2 Proposed model ........ 20

3.2.1 Data Collection ........ 21

3.2.2 Data Pre-processing ........ 22

3.2.3 Dictionary Generation ........ 23

3.2.4 Bag-of-Words Generation ........ 24

3.2.5 Topic Modelling ........ 24

3.2.6 Latent Semantic Analysis (LSA) ........ 24

3.2.7 Latent Dirichlet Allocation (LDA) ........ 25

4 RESULT AND DISCUSSION ........ 30

4.1 Introduction ........ 30

4.2 Results ........ 30

4.2.1 Results for 2015 ........ 30

4.2.2 Results for 2016 ........ 32

4.2.3 Results for 2017 ........ 34

4.2.4 Results for 2018 ........ 35

4.2.5 Results for 2019 ........ 37

4.3 Discussion ........ 39

5 CONCLUSION AND RECOMMENDATION ........ 40

5.1 CONCLUSION ........ 40

5.2 RECOMMENDATION ........ 40

References ........ 41

Appendix ........ 44

# List of Figures and Tables

Figure 1-1 Data Mining Process .......... 2
Figure 3-1 Proposed model for Twitter Data Mining .......... 20
Figure 3-2 Tweets retrieved from @nighealthwatch .......... 21
Figure 3-3 Data before pre-processing.......... 23
Figure 3-4 Data after pre-processing.......... 23
Figure 3-5 Generated Dictionary .......... 23
Figure 3-6 Coherence score over a range of topic numbers.......... 26
Figure 3-7 U-Mass Coherence Scores.......... 28
Figure 3-8 C_V Coherence Scores.......... 29
Figure 4-1 Coherence score for various number of topics 2015.......... 30
Table 4.1 Topic Table for Year 2015 Data .......... 31
Figure 4-2 Coherence Scores for various number of topics 2016.......... 32
Table 4.2 Topic Table for Year 2016.......... 33
Figure 4.3 Coherence score for various number of topics 2017 .......... 34
Table 4.3 Topic Table for Year 2017.......... 35
Figure 4.4 Coherence Score for various number of Topics 2018 .......... 36
Table 4.4 Topic Table for Year 2018.......... 37
Figure 4.5 Coherence score for various number of topics 2019 .......... 38
Table 4.5 Topic Table for Year 2019.......... 38

# 1 INTRODUCTION

## 1.1 Introduction

There has been an exponential increase in the availability of data over the past years. According to Hal Varian, Chief Economist at Google, “*Between the dawn of civilization and 2003, we only created five exabytes; now we're creating that amount every two days. By 2020, that figure is predicted to sit at 53 zettabytes (53 trillion gigabytes) -- an increase of 50 times*." While we generate 2.5 quintillion bytes of data every day, 90% of the worlds data has been created in the past two years alone (Winans et al., 2017). These data are generated from the internet, social media, IoT, through communication, digital photos, videos and services. With this increase and availability of data comes the question of what we can do with it because the data growth phenomenon continues. With smart phones and internet getting more affordable and available, the number of social media users is on the rise; this again shows an increase in data generation and availability. Every minute; Google conducts 3,877,140 searches, 49,380 users post on Instagram, 4,333,560 videos are streamed on YouTube and 473,400 tweets are sent on Twitter (Data Never Sleeps 6.0, 2018). Based on this statistic, the question once again is how can available data be used? A lot of these data come in unstructured and text format and are mined using special techniques like information retrieval, clustering, text summarization and topic modelling. Insights in politics, business, entertainment and health can be derived from the loads of data available by applying topic modelling technique.

## 1.2 Data Mining

Data mining encompasses numerous techniques and processes. It can be defined as the process of gaining meaningful insight and patterns from a large data set. Various forms of data (text, numeric, time series, structured unstructured etc.) require different techniques. The data mining pipeline typically has 3 phases(Aggarwal, 2015).

### 1.2.1 Data collection

Data collection is the phase of gathering the right data to accomplish a task at hand. This most times implies gathering data from various sources like surveys and questionnaires, sensors, web scrapping, etc.; as the required data might not be in one place. This phase is critical because the quality of the data gathered affects the result of the entire mining process. The data must be relevant to the task; as the old saying garbage in garbage out also applies in data mining.

### 1.2.2 Feature extraction and data cleaning

Real world data does not come in very good shape most times. There could be missing data, unrealistic data like negative ages, poorly scaled data like salaries ranging from N100 to N100,000,000 and so on. Hence there is a need to clean up the collected data. Also, not features / characteristics of the data might be necessary for the mining process. Hence after the data has been collected and cleaned, we must choose the needed features for the mining process.

### 1.2.3 Analytical processing and algorithms

At this phase our data is set for mining and depending on the task at hand, the appropriate process(es) and algorithm(s) are chosen and used on the data. Fig. 1-1 below gives an overview of the data mining process.

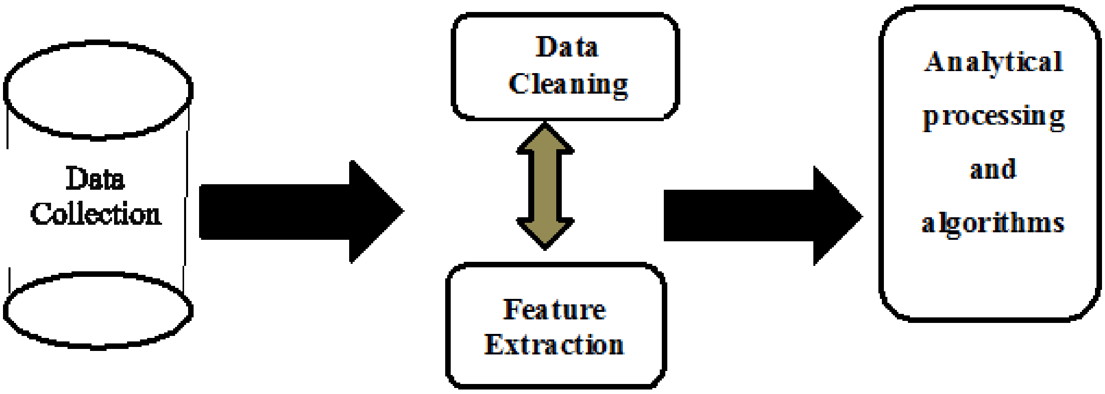


*Figure 1-1 Data Mining Process*

## 1.3 Text Mining

Text mining also known as text analytics is a streamline of data mining. In text mining, we are looking for insights from large text data set which often comes unstructured. Different text mining tasks require different mining techniques. Text mining techniques includes text categorization, text clustering, entity extraction, sentiment analysis, document summarization, topic modelling etc.

The text mining cycle is like the data mining cycle but includes a phase of 'data structuring' after data collection as the techniques cannot be applied directly on the unstructured text data.

## 1.4 Topic Modelling

Given a text data set; usually a collection of documents, one common task is to derive the topics in that data. Topic modelling is the process of applying statistical models (topic models) to extract the hidden / latent topics in the data. These models work by getting the hidden patterns in the document collection. Existing topic models include:

i. Latent Semantic Analysis (LSA)

ii. Probabilistic Latent Semantic Analysis (PLSA)

iii. Latent Dirichlet Allocation (LDA)

iv. Correlated Topic Model (CTM)

v. Explicit semantic analysis

vi. Hierarchical Dirichlet process

vii. Non-negative matrix factorization

## 1.5 Applications of Topic Modelling

- In information retrieval (IR), topic models are used for Smoothing language models, query expansion, search personalization (Boyd-Graber, Jordan & Hu, Yuening & Mimno, 2017).

- Topic models are used to track topical changes in various fields influenced by historical events through considering newspapers, historical records, and historical scholarly journals(Boyd-Graber, Jordan & Hu, Yuening & Mimno, 2017)
- In the literary world, topic models are used to analyse the creative, diverse oeuvre of authors and the emotions and thoughts of fictional characters(Boyd-Graber, Jordan & Hu, Yuening & Mimno, 2017).
- With the lots of online discussions across social media platforms, topic models can aid companies understand their customers, politicians target voters and researchers the impact of social media on people's everyday life through unlocking the emotion and hidden factions often present in online discussions(Boyd-Graber, Jordan & Hu, Yuening & Mimno, 2017).

## 1.6 Problem Statement

The boom of the Internet and social media (in particular Twitter) is still young in Nigeria. With Nigeria being the second most twitting country in Africa(Portland Communications, 2016) and having 6 million active users (Terragon Group, 2018), lots of data are generated daily via this media daily from which vital information can be retrieved to better decision making. We seek to answer the question: what meaningful insights can be gained from this data through topic modelling? The outcome of this research will spark local utilization of data through topic modelling and further research in this area as it is ever growing.

## 1.7 Aim and Objectives

The aim of this work is to find out the top health topics in Nigeria over the past few years by applying topic modelling on Nigerian health twitter data to demonstrate the potential of twitter mining.

As a developing country which still battles with numerous health issues, we seek to find out the top health topics in Nigeria over the past few years by achieving the following objectives:

- Collect the tweets from major databases
- Use LDA to find out topics for a year
- Use LSA to find out topics for a year
- Compare the results of LDA and LSA to choose the better.
- Perform twitter mining with the better topic model for data from 2015 – 2019.

## 1.8 Methodology

We follow the following steps in achieving our objectives:

1. The needed data is not domiciled in a location and must thus be collected. Top official health twitter accounts are identified, and their tweets collected through the GetOldTweets3 python library. Since tweets are usually very short (280 characters) we aggregate all tweets from an account to make a document.
2. The data collected is noisy and unstructured hence we next clean it and put it into a structure fit for topic modelling. This step is carried out repeatedly.
3. LDA is next performed on the data to get latent topics.

Python language is used for this thesis on google Collaboratory which provides GPU on the cloud for faster computation.

In the next chapter, we summarize literatures on what has been done in the field of twitter mining. The methodology used is detailed in chapter 3 and the result reported in chapter 4. Chapter 5 states conclusion and suggest further work to be done.

# 2 LITERATURE REVIEW

## 2.1 Introduction

In 2006, Clive Humby a UK mathematician said “*Data is the new oil. It’s valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc to create a valuable entity that drives profitable activity; so, must data be broken down, analyzed for it to have value*”.

Social media like Facebook, Instagram, Snapchat, Twitter, WhatsApp, etc. generate a great percentage of this new oil. Twitter which is a free platform for individuals and organizations to broadcast information generates around 200 billion tweets yearly (“Twitter Usage Statistics - Internet Live Stats,” n.d.). Since data is considered the new oil and it is in abundance, the question is what has been done with it? According to literature, this available huge data has been utilized to solve different problems.

In this chapter, we review these works that have been done with twitter data according to literature chronologically from 2014 till date; this also demonstrated the relevance of this research.

## 2.2 Basic Terminologies

In order to understand recent studies, we define basic terminologies.

### 2.2.1 Term/Token

These are the building blocks of documents. It refers to words, phrases, symbols, or other meaningful elements in a document(Andrius Velykis, 2018).

### 2.2.2 Document

A document is a piece of text regardless of its size. It could be as short as single sentence or as large as hundred pages(Provost & Fawcett, 2016).

### 2.2.3 Corpus

This refers to a collection of documents which could be related (often the case) or not.

### 2.2.4 Bag of words

A corpus in its form cannot be fed into text mining algorithms and must be transformed to a data form suitable for these algorithms. Bag of words is a text representation in which each document is treated as just a collection of individual words which are treated as potentially important keywords while ignoring grammar, word order, sentence structure and most times punctuations (Provost & Fawcett, 2016).

### 2.2.5 Term frequency (TF)

TF measures how frequent a word is in a document. It is the number of times a word occurs in a document. Depending on the mining task, a term should not be too rare or too common. Although term frequency is an easy measurement of a term's prevalence, by considering raw counts, terms which are less important and occur often in documents like "the" or "a" will be emphasized more than other keywords.

TF could be the raw count of a term in a document or the raw count normalized by dividing by the total number of terms in the document, defined by

I. $TF(f,d) = \text{number of times term t appears in a document d}$

II. $TF(f,d) =$

$$(\text{number of times term t appears in a document d}) \;/\; (\text{total number of terms in the document d})$$

### 2.2.6 Inverse Document Frequency (IDF)

IDF mitigates the challenge of term frequency which could allow a word which is less important to rank higher than more important words in a search since TF considers all terms to be equally important. IDF measures the importance of a term by weighing down frequent terms and scaling up rare terms. IDF is defined mathematically as follows:

$$IDF(t,D) = log_e(\text{Total number of documents D} \;/\; \text{Number of documents with term t in it})$$

### 2.2.7 Term Frequency–Inverse Document Frequency (TF-IDF)

TF-IDF measures the relevance of a term into a given document in a corpus. TF-IDF can also be used to remove stop words during data pre-processing for text mining. It is defined as the product of TF and IDF, mathematically:

$$TF - IDF(t,d,D) \ = \ TF(f,D) * IDF(t,D)$$

TF, IDF and TF-IDF are term weighing schemes in NLP.

### 2.2.8 Document- Term matrix

Document – Term matrix corresponds to the frequency of terms in a corpus. The rows correspond to respective documents while the columns correspond to the terms in the documents. The fields could be TF, IDF or TF-IDF.

### 2.2.9 Document-Topic matrix

Document – Topic matrix is a mathematical matrix in which the rows correspond to the documents in the corpus while columns correspond to the topics in the corpus. The fields represent the probability of a topic in a document.

### 2.2.10 Topic-Word matrix

Topic-word matrix is a mathematical matrix in which the rows correspond to the topics in the corpus while columns correspond to the terms in the corpus. The fields represent the probability of a word in a topic.

## 2.3 Topic modelling

Topic modelling is a type of statistical modelling for discovering the latent "***topics***" that occur in a corpus. Topic models are used to summarize data, especially text data in terms of a small set of latent variables which could be referred to as topics. They are also tools for dimensionality reduction. The following are examples of topic models.

### 2.3.1 Latent Semantic Analysis (LSA)

LSA is a technique in NLP for analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms (Dumais, 2005). It uses Singular Value Decomposition (SVD) on the Document-Term matrix of a corpus whose field are usually TF-IDF of terms in documents to identify a linear subspace in the space of TF-IDF features that captures most of the variance in the corpus. SVD is based on the linear algebra theory which says any matrix $A_{m,n}$ can be written as

$A_{m,n} = U_{m,m} * S_{m,n} * V_{n,n}$. Where $A_{m,n}$ is the Document-Term matrix. Upon applying SVD, the resulting $U_{m,m}$ is the Document-Topic matrix while $V_{n,n}$ is the Document-Topic matrix. Although LSA is efficient and quick to use, it lacks interpretable embedding's, requires a really large data set for accuracy and has a less efficient representation(Xu, 2018).

### 2.3.2 Probabilistic Latent Semantic Analysis (LSI)

LSI also known as Aspect model is an alternative to PLA which uses a probabilistic model instead of SVD. It expresses the data in terms of observed (documents and terms) and latent(topics) variables. The crux of PLSA is to find a probabilistic model with latent variable (topics) that can generate our data as in the Document- Term matrix (Xu, 2018). The Aspect model is defined thus:

$$P(d,w) = P(d) \sum P\langle w|z\rangle P\langle z|d\rangle$$

$P(d)$ --- probability of a document

$\sum P\langle w|z\rangle$ --- probability of a word given a document

$P\langle z|d\rangle$ --- probability of a topic given a document

All though PLSI model out performs LSI model, the number of parameters to be estimated grows linearly with the size of training documents, it is prone to overfitting and it is not a well-defined generative model (Blei, Ng, & Jordan, 2003).

### 2.3.3 Latent Dirichlet Allocation (LDA)

LDA is a three- level hierarchical Bayesian model for a collection discrete data such as a corpus (Blei et al., 2003). Simply put, documents are represented as random mixtures over latent topics and each topic is in turn characterized by a distribution over words (Blei et al., 2003).

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by(Blei et al., 2003):

$$p(\theta, z, w|\alpha, \beta) = p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta)$$

### 2.3.4 Lda2Vec

Lda2Vec is an extension of LDA topic modelling and skip-gram word2vec word embedding which jointly learns word, document and topic vectors (Xu, 2018).

The data set we use for this project is large and unlabeled. Going through the thousands of tweets to label them and apply a supervised is a possible but highly tedious and time-consuming task. Therefore, we are using LDA topic modelling in this project to identify the topics in our dataset.

## 2.4 Review of Literatures

When faced with a huge number of tweets for analysis, one of the fundamental challenges is aggregation. We have tweets from various users, possibly discussing different subject matters, how do we arrange them to arrive at some form of order? To this end, some aggregation methods have been proposed including the following:

i. **Basic Scheme (**Unpooled**):** is the base technique in which each tweet is considered a document and the topic model is applied directly to the collection of tweets (documents). Since LDA does not perform well on document with short length(Zhao et al., 2011), this technique yields a poor result.

ii. **Author-wise Pooling** is a method that aggregates all tweets from a username/account into a document. This is a standard way of aggregating twitter data(Mehrotra, Sanner, Buntine, & Xie, 2013).

iii. **Burst-score wise Pooling:** is an pooling technique in which tweets are aggregated according to the terms having a high burst score they contain(Mehrotra et al., 2013).

iv. **Temporal Pooling:** is a technique in which tweets gathered within a time frame generally about a subject are randomly aggregated into documents (Mehrotra et al., 2013).

v. **Hashtag-based Pooling:** aggregates tweets with similar hashtag into a document and If a tweet has multiple hashtags, the tweets is put in each document for each hashtag(Mehrotra et al., 2013). Since a hashtag in general summarizes the message in a tweet, aggregating by hashtags results in most related tweets falling under the same document and yielding better topics after LDA as shown by experiments by (Mehrotra et al., 2013).

vi. Although aggregating by hashtags works successfully well, the challenge arises when tweets do not contain hashtags which often happens. This is handled by **aggregating by named entities and word collocations** proposed two years later by (Samarawickrama, Karunasekera, & Harwood, 2015). Named entities refer to names of persons, organization, location, etc. Although this scheme yields even better results than the afore discussed schemes after experiments (Samarawickrama et al., 2015), the named entity recognizer StanfordNER is a good named entity recognizers for English and a few other languages like German, Chinese and Spanish ("The Stanford Natural Language Processing Group," n.d.). This implies that in the Nigerian context this method will no work well as we do not have a good Nigerian name entity recognizer as of the time of this research.

We evidently live in the era of big data which defiles traditional data analytics methodologies. Hence the challenges of analytics, pattern recognition, visualization, etc. (Anjaria & Guddeti, 2014) of big data which are mostly unstructured and textual such as twitter data. One of the end goals of twitter data analytics is prediction. In commerce, we want to predict the likelihood of a commodity being purchased by potential customers, in politics we want to predict the likely winner of an upcoming election; all based on the available twitter data. This is possible because of the available and algorithms. With Machine Learning algorithms like Support Vector Machines (SVM), Naïve Bayes, Maximum Entropy and even Artificial Neural Networks predictions can be made on different subject matters using twitter data. However, the question is how do we improve the accuracy of these predictions? (Anjaria & Guddeti, 2014) proposed the addition of some information like the sentiment of the tweet and even more personal information such as age, educational background, employment status, economic criterion, rural & urban and social development index to the twitter data for the improvement of predictions. Although the experiment conducted by (Anjaria & Guddeti, 2014) demonstrated the improvement in predictions by the additional information, these information are not readily available especially the personal information.

On an individual level, it is useful to ask the question; what are microblog users like twitter interested in? For targeted advertising, friend recommendation and people profiling, knowing the interests of users is important. Topic models are useful for discovering the interests of microblog users. Although traditional topic models like LDA can be used to find the interest of users (topics) on microblogs like twitter, (He, Jia, Han, & Ding, 2014) proposed a novel topic model called User-Topic model to discover the interest of microblog users. According to the experiment carried out by (He et al., 2014) on data from Sina microblogs which at the time of experiment had the 140-character limit same as twitter, the proposed model effectively discovered users interest in microblogs. The User-Topic model is based on LDA. In it, the interests of each user are divided into two parts by different ways to generate the original interest and retweet interest. In the end, the original interest and retweet interest are combined to compute the interest words for users.

Be it tweets pertaining to an individual, organization or subject matter; after deriving the topics contained in the twitter dataset as probabilities of words using topic modelling like LDA, tweets can be labelled to a particular topic with the aid of the topic words and similarity schemes like cosine similarity (Samarawickrama et al., 2015).

Beyond knowing users' interests, discovering their emotions through their tweets is also important. Is the user happy, sad or just neutral based on her tweets? This is called sentiment analysis. The availability of big amounts of tweet data has made sentiment analysis of tweets possible. It is a typical classification problem that is useful to organizations for monitoring the satisfaction of customers using through their feedback on Tweeter. (Tripathi, Vishwakarma, & Lala, 2015) used two different classifiers; Naïve Bayes classifier and K-Nearest Neighbor classifier to predict the sentiment of a tweet as happy, sad or neutral. The bottle neck of this text mining operation is that it requires labelled data. The human effort required to label the data increases as the data increases and can become a huge challenge with very big data. However, K-NN classifier predicted the emotions of tweets better according to the experiments conducted by (Tripathi et al., 2015). The performance of the classifiers was measured by their precision and recall.

Happy, sad or neutral are not the only sentiments that can be measured from tweets. Other emotions, behaviors, attitudes, tones and awareness are also measured through tweets (Nirmala, Roopa, & Kumar, 2015). This is because sentiment analysis has numerous applications from public opinion and customer feedback analysis to product analysis and market research and analysis. One application with the available twitter data and algorithms for sentiment analysis is to discover the unemployment rate for a period of time by analyzing the results from sentiment extraction from the tweets. (Nirmala et al., 2015) attempted to do this by experimenting on public twitter data related to unemployment. To mitigate the labelling challenge, the task of labelling tweets as positive or negative was automated by scoring tweets based on dictionaries of positive and negative terms. Each tweet was given score generated by $\text{positive score} - \text{negative score}$ where positive score and negative score were the number of positive and negative words in the tweets respectively.

Score < 0 implies an overall positive opinion for the tweet

Score > 0 implies an overall negative opinion for the tweet

Score = 0 implies a neural positive opinion for the tweet

Even though this approach saves time and human effort, it is highly prone to erroneous classification due to the complexity of human language. A tweet like *“the film hot die”* will be classified as negative due to presence of more negative terms. However, in the Nigerian context the tweet implies the film was very interesting which is positive. By applying sentiment analysis and text mining techniques, (Nirmala et al., 2015) demonstrated the strong correlation between negative feelings in public opinions (negative tweets) and the unemployment rates. The experiment on the mined Twitter data showed that at the time of the research, not many jobs were available and there was an unemployment crisis.

In text mining, we want more robust techniques and tools, to be able to achieve more with less. Like the discovery of topics with the sentiments associated from a twitter data set. (Ahuja, Wei, & Carley, 2016) also proposed two probabilistic graphical models for sentiment analysis and topic modelling. The proposed models for topic modelling are Microblog Topic Model (MTM) and Microblog Sentiment Topic Model (MSTM). MSTM discovers topics as well as sentiments associated with the topics. MTM and MSTM are generative models that assume each social media post like tweets stem from a single topic and model words and hashtags (which is considered a special meta token) separately. The results of the experiments carried by (Ahuja et al., 2016) show that the models were effective for the purpose for which they were designed i.e. discovering latent topics and their sentiment from social media data like Twitter data. However, the assumption that each tweet stem from a single topic will many times be wrong because tweets often stem from various topics as indicated by accompanying varying hashtags.

So far, we have highlighted the beauty of having and mining a twitter data set. However, some challenges that exist with mining twitter data must be highlighted for balance. Firstly, tweets derived from twitter come with lots of data that cannot be handled easily like emojis (Wolny, 2016). Next, tweets are many a times sparse which is disadvantageous to text mining algorithms (Wolny, 2016). Finally, more that 40% of tweets are informal in nature making it difficult to discover topics (Wolny, 2016).

Despite these challenges, heavy advancement has been made in the mining of twitter data as above stated and more. People profiling can be performed through twitter as twitter users provide their name, username, email address or phone number when creating accounts. Other optional information like a short biography, location website, profile picture or date of birth are provided. These information can be obtained from the twitter REST API for collecting users' information (Wolny, 2016) for people profiling. With keys like 'coordinates', 'geo' and 'place' in a tweet that determine the tweet author's location (Wolny, 2016), insights can be gained on the correlation between the tweets and society via location.

Insights from mining twitter data considering geographical location and time stamps are useful to planners, marketers and researchers as it provides useful information about activities and opinions across time and space (Lansley & Longley, 2016).

(Lansley & Longley, 2016) used an unsupervised learning algorithm (LDA) to classify geo-tagged Tweets from inner London recorded during typical weekdays for the year 2013 into a small number of groups. The process yielded 20 distinct and interpretative topic groupings that represent key types of Tweets like activity description, conversations and app check-ins. The experiments by (Lansley & Longley, 2016) demonstrated that Twitter users do not Tweet evenly across space and Tweets are affected by land-use (geography) and activities like fashion, shopping, shows, nightlife, etc.

Being able to get real time information from Twitter for decision making is of great importance since lots of information are disseminated on Twitter before traditional media like newspapers or television stations. This presents the possibility of detecting breaking news as they emerge. To achieve this, we need to be able to differentiate new-worthy Tweets from normal public opinions. Instead of maintaining a list of all keywords that represent news, topic modelling can be used to discover new-related Tweets even without a list of keywords that constitute news (Wold, Vikre, Gulla, Özgöbek, & Su, 2016). In (Wold et al., 2016) four topic models where compared through experiments.

1. Latent Dirichlet Allocation (LDA) considering each tweet as a document
2. Latent Dirichlet Allocation (LDA) aggregating tweets per author
3. Hierarchical Dirichlet Process (HDP) considering each tweet as a document
4. Hierarchical Dirichlet Process (HDP) aggregating tweets per author

Although the results of the experiment showed that Latent Dirichlet Allocation (LDA) aggregating tweets per author outperformed other three models, it also showed that this model alone is not likely sufficient for the task of detecting breaking news. This is because the shortness and ambiguity of tweets make it difficult to generate statistical models of the required precision for the task of breaking news detection.

Besides using topic models like LDA to discover topics from a Twitter data set, other unsupervised learning algorithms like K-means and CLOPE clustering can applied for trending topic detection. In (Sapul, Aung, & Jiamthapthaksin, 2017) the results of using K-means, CLOPE clustering and LDA topic modelling algorithms for detecting topics on tweets were compared. The results showed that CLOPE algorithm discovered more topics compared to K-means and LDA. This however could be due to the fact that CLOPE algorithm automatically generates the number of clusters where as LDA and K-means requires that the number of clusters (topics) be set manually which if not optimally chosen would yield a poor result. (Sapul et al., 2017) also showed that including more terms like hashtags results in better topic discovery from the Twitter dataset upon applying theses algorithms.

In (Zulfikar, Irfan, Alam, & Indra, 2017) Naive Bayes Classifier, Nearest Neighbour, and Decision Tree were compared for their effectiveness in Indonesian Swear Words detection on Twitter data. Further demonstrating insights that can be derived from Twitter data. Nearest Neighbor emerged more accurate the Nearest Neighbor, and Decision Tree for this task of classifying a tweet as dirty or clean in terms of swear word content.

In recent years, Deep learning algorithms have been used for better results in classification tasks. It has also been used to perform sentiment analysis on Twitter data. In (Ramadhani & Goo, 2017) deep learning methods were used for Twitter sentiment analysis. The deep neural network (DNN) gave a better result of an average accuracy of 76.24% compared to the result of the Multilayer perceptron which gave an average accuracy of 60%.

Even though Twitter provides a lot of data that can be mined to gain useful insights for decision making, sometimes events broadcasted on Twitter are false. This is a concern as the spread of false information has the potential of harming individuals and societies. Hence an important question is raised; how do we make a distinction of true and false events on Twitter?

In 2018 a lot of work was done on Tweeter mining. (Hassan, 2018) developed a text mining approach to respond to these concerns automatically. To do this, (Hassan, 2018) used annotated event Tweets from CREDBANK on several machine learning algorithms viz Support Vector Machine (SVM), Decision Trees (DT), Random Forest (RF) and Naïve Bayes (NB) on WEKA. WEKA is an open source data mining software. The experiment results in (Hassan, 2018) showed that Decision Tree had the overall accuracy, 82.26%. This is good and a well-tested system based on this for filtering Tweets based on the credibility of content will play a huge role in Tweet mining as the output of any Tweet mining is dependent on the Tweets inputted.

(Yang & Zhang, 2018) explored and demonstrated the principles, theories and applications of LDA topic modelling and sentiment analysis on Twitter plain text in

English using R programming language. (Yang & Zhang, 2018) derived 15 topics from their experiment and discovered 61.5% of the data set contained positive terms while 38.5% had negative terms.

Using LDA, (Hidayatullah, Pembrani, Kurniawan, Akbar, & Pranata, 2018) discovered the topics of tweets about football news on Bahasa Indonesia further demonstrating the applications of Twitter mining. To do this, tweets were retrieved from Twitter API and the python GetOldTweets library. Retrieved tweets were the pre-processed and used for LDA topic modelling. (Hidayatullah et al., 2018) discovered 10 topics including El-classico, Serie A Italy and World Cup.

Twitter mining has also benefited businesses as it enables companies to discover the sentiments of customers towards a product or service. (Halibas, Shaffi, & Mohamed, 2018) discussed the technical and business perspectives of Twitter analysis. In (Halibas et al., 2018), experiments were conducted for both clustering (K-means) and classification (Decision Tree) on customer comments from Twitter on a popular food brand. From the results of the classification experiment, the model predicted 85% negative comments and 15% positive comments. The results of the clustering experiments confirmed the result of the classification. This raises a red flag which the company must investigate and demonstrates the role of Twitter mining in the decision making of businesses.

In the health sector, Twitter mining has also been useful. (Asghari, Sierra-Sosa, & Elmaghraby, 2019) presented a 5-layer adaptive Twitter analysis system made up of both supervised and unsupervised learning algorithms to track health trends on social media. The system uses LDA to label each tweet by identifying patterns and Convolutional Neutral Networks (CNN) together with word2vec model for classification. The system performed very well with an accuracy of 83.34%, precision of 83%, recall of 84% and F-Score of 83.8%. The system discovered trending health topics such as Diabetics, Digital health and Care.

Most recently in 2019, (Ramanathan & Meyyappan, 2019) recommended a new sentiment analysis method based on common sense knowledge (Prior to this proposal, Lexicon based approach and machine learning approach were the two most popular approaches). This required them creating their own Oman foundation ontology since experiments were conducted on Tweets about Oman tourism. Based on the results of the experiments in (Ramanathan & Meyyappan, 2019), the proposed method of sentimental analysis including conceptual semantic sentiment analysis significantly improves the performance of sentiment analysis.

## 2.5 Conclusion

The usefulness of Twitter data has only been scratched on the surface so far. This is because we can get information on nearly every topic of interest from Twitter; from entertainment to politics and even religion.

Since Twitter has been available for the past thirteen years, it contains historical data on different topics and hence can be used for a track historic trends in different fields.

In this work, we will be mining Nigerian Health Twitter data for the past five years to derive different topic and possible trends using topic modelling.

# 3 METHODOLOGY

## 3.1 Introduction

From the available Twitter data, we seek to gain insights that are not obviously visible to us due to our limitations as humans to process big data. However, since the data gotten from Twitter is unlabelled, we will be using topic modelling which is an unsupervised algorithm for our experiments.

Each experiment requires five steps of data collection, data preprocessing, dictionary generation, Bag-of-Words generation and topic modelling. We begin by comparing two topic models Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) in other to choose the better for our experiment on Nigerian health Twitter data from 2015 till date. Python programming language was used for this work.

## 3.2 Proposed model

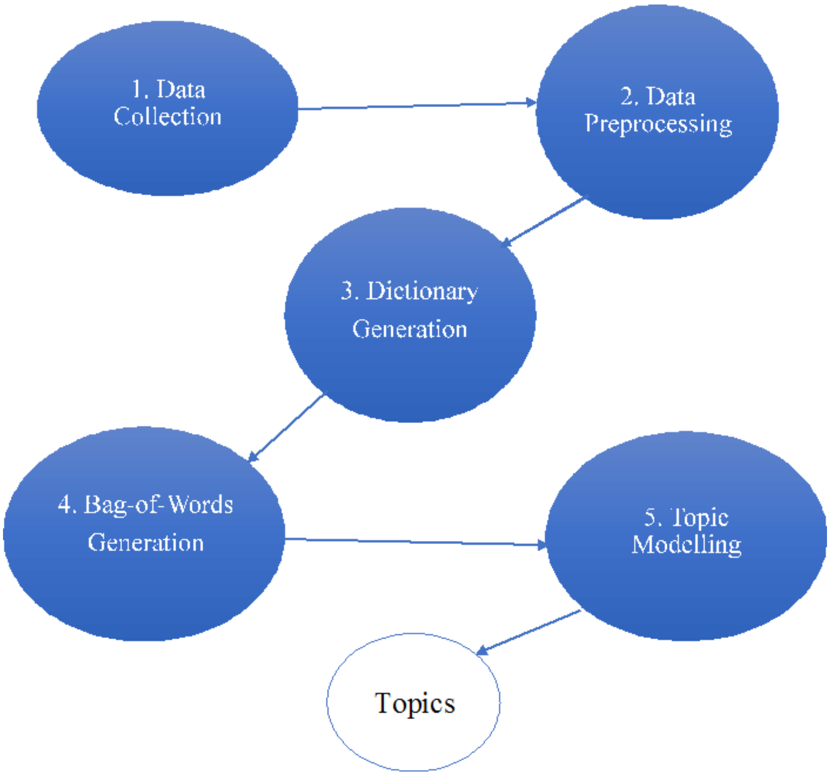


*Figure 3-1 Proposed model for Twitter Data Mining*

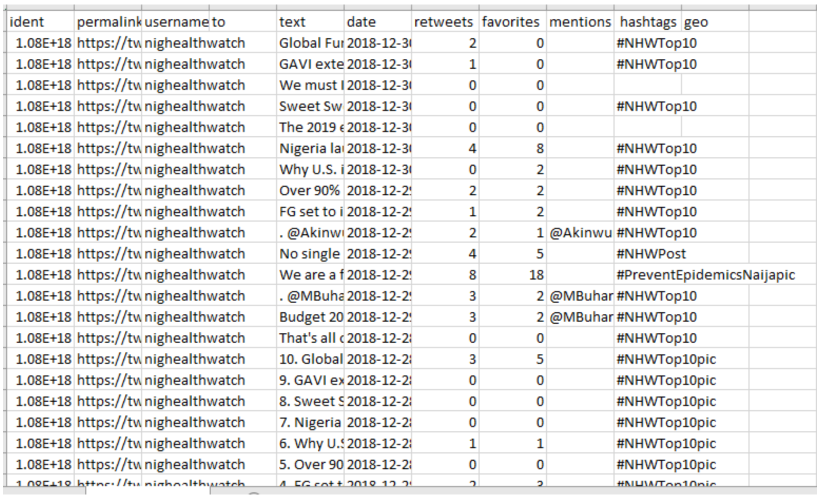
Fig. 3-1 gives a pictorial summary of our proposed model pipeline for twitter data mining.

### 3.2.1 Data Collection

Our focus is on Twitter text data. Tweets are messages passed on the Twitter platform. A Tweet can be an original post by a user, a direct message to another user or a share of another user’s tweet is referred to as a retweet. Tweets can contain non-ASCII characters like emojis and images. Although tweets come with a lot of meta data like location, permalink, username, to, text, date, retweets, favourites, mentions and more, we are strictly interested in the texts for this work. Tweets have a limit of 280 characters as of the time of writing.

In other to gather data for our experiments, we Identified verified Twitter accounts that post Nigerian health news. We identified eleven twitter handles viz: @nighealthwatch, @nmanigeria, @Fmohnigeria, @NphcdaNG, @EpiAFRIC, @APINNigeria, @W4HNigeria, @SFHNigeria, @NCDCgov, @wharc_nigeria and @WHONigeria.

To retrieve the tweets from these accounts we used the GetOldTweets Python library. We used this library as against twitter streaming API and Twitter Archiver because it allows us to retrieve very old tweets from years ago for free unlike the twitter streaming API which allows us to stream tweets from only few weeks back and Twitter Archiver which requires payment for full functionality. The retrieved tweets were retrieved into CSV files.



| ident | permalink | username | to | text | date | retweets | favorites | mentions | hashtags | geo |
|---|---|---|---|---|---|---|---|---|---|---|
| 1.08E+18 | https://tw | nighealthwatch |  | Global Fu | 2018-12-3 | 2 | 0 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | GAVI exte | 2018-12-3 | 1 | 0 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | We must I | 2018-12-3 | 0 | 0 |  |  |  |
| 1.08E+18 | https://tw | nighealthwatch |  | Sweet Sw | 2018-12-3 | 0 | 0 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | The 2019 | 2018-12-3 | 0 | 0 |  |  |  |
| 1.08E+18 | https://tw | nighealthwatch |  | Nigeria la | 2018-12-3 | 4 | 8 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | Why U.S. i | 2018-12-3 | 0 | 2 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | Over 90% | 2018-12-2 | 2 | 2 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | FG set to i | 2018-12-2 | 1 | 2 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | . @Akinw | 2018-12-2 | 2 | 1 | @Akinwu | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | No single | 2018-12-2 | 4 | 5 |  | #NHWPost |  |
| 1.08E+18 | https://tw | nighealthwatch |  | We are a f | 2018-12-2 | 8 | 18 |  | #PreventEpidemicsNaijapic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | . @MBuha | 2018-12-2 | 3 | 2 | @MBuhar | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | Budget 20 | 2018-12-2 | 3 | 2 | @MBuhar | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | That's all | 2018-12-2 | 0 | 0 |  | #NHWTop10 |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 10. Global | 2018-12-2 | 3 | 5 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 9. GAVI ex | 2018-12-2 | 0 | 0 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 8. Sweet S | 2018-12-2 | 0 | 0 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 7. Nigeria | 2018-12-2 | 0 | 0 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 6. Why U.S | 2018-12-2 | 1 | 1 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 5. Over 90 | 2018-12-2 | 0 | 0 |  | #NHWTop10pic |  |
| 1.08E+18 | https://tw | nighealthwatch |  | 4. FG set t | 2018-12-2 | 2 | 3 |  | #NHWTop10pic |  |

*Figure 3-2 Tweets retrieved from @nighealthwatch*

After retrieving our data as CSV files, we transform them into dataframes using Pandas Python library as shown in Fig 3-2. Pandas is a data manipulation and analysis library in Python.

Since we are interested in only the text of the tweets, we create new dataframes which are slices of only the text of the tweets.

### 3.2.2 Data Pre-processing

The pre-processing stage is the crucial stage where noise is reduced from our dataset. Pre-processing our dataset involves the following steps:

i. Convert to lowercase: we first convert all text to lowercase. This is so that same words written in different cases will not be considered as being different.

ii. Remove URLs: URLs are not words with meanings and hence will not be useful for our experiments. In other to remove URLs from our dataset, we used regular expression to define and identify the patterns of URLs and replace them with a ““.

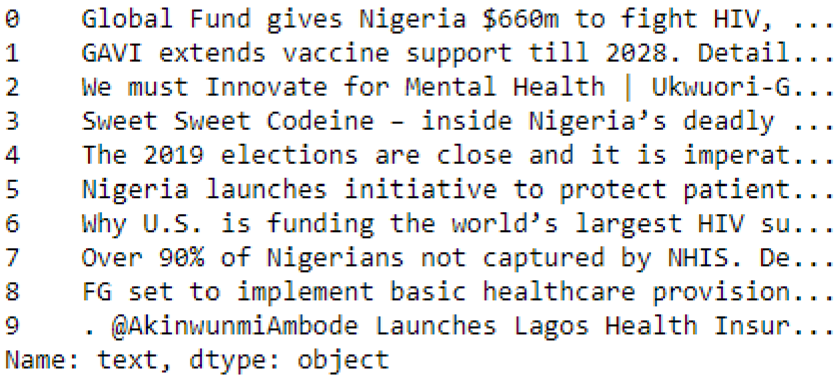
iii. Lemmatization: sometimes a word is used in its various forms. For instance, the word *sing* can be used as *sang, sung, singing* all of which come from the same root word *sing*. Lemmatization is the process of deriving the root of words by removing prefixes and suffixes. We used the lemmatization module in Python’s NLTK library to perform lemmatization on our data.

iv. Digits removal: for our experiments we are interested in only words. So, using regular expressions we replace every occurrence of digits with “”.

v. Tokenization: this is the process of breaking down our data into single words. “ “ is used as the delimiter for tokenization.

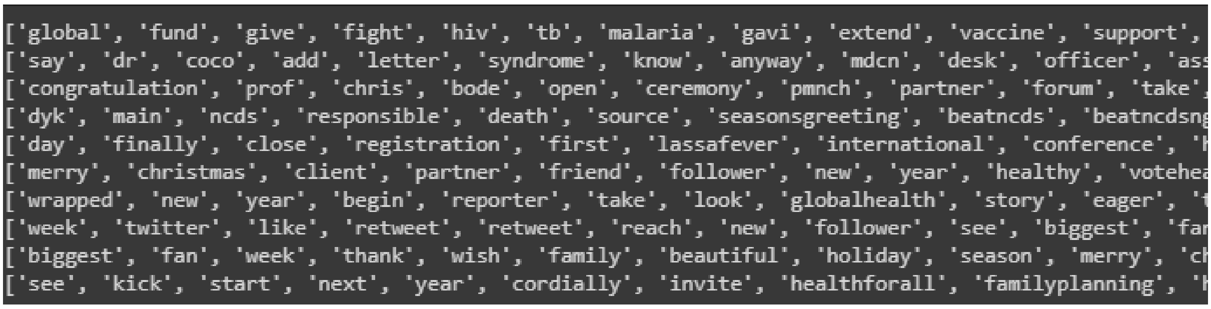
vi. Stopword removal: there are words that occur often in texts and most times do not convey the core message of the text. These are referred to as stopwords. This are words like "a," "and," "but," "how," "or," and "what.".

Fig. 3-3 and Fig 3-4 show our data before and after pre-processing respectively.



```
0    Global Fund gives Nigeria $660m to fight HIV, ...
1    GAVI extends vaccine support till 2028. Detail...
2    We must Innovate for Mental Health | Ukwuori-G...
3    Sweet Sweet Codeine – inside Nigeria’s deadly ...
4    The 2019 elections are close and it is imperat...
5    Nigeria launches initiative to protect patient...
6    Why U.S. is funding the world’s largest HIV su...
7    Over 90% of Nigerians not captured by NHIS. De...
8    FG set to implement basic healthcare provision...
9    . @AkinwunmiAmbode Launches Lagos Health Insur...
Name: text, dtype: object
```

*Figure 3-3 Data before pre-processing*

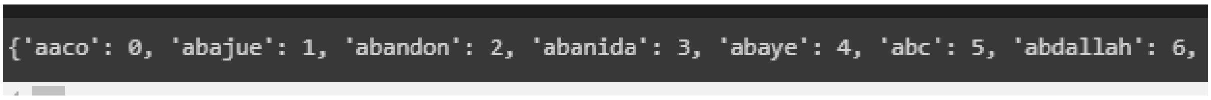


```
['global', 'fund', 'give', 'fight', 'hiv', 'tb', 'malaria', 'gavi', 'extend', 'vaccine', 'support',
['say', 'dr', 'coco', 'add', 'letter', 'syndrome', 'know', 'anyway', 'mdcn', 'desk', 'officer', 'as
['congratulation', 'prof', 'chris', 'bode', 'open', 'ceremony', 'pmnch', 'partner', 'forum', 'take',
['dyk', 'main', 'ncds', 'responsible', 'death', 'source', 'seasonsgreeting', 'beatncds', 'beatncdsn
['day', 'finally', 'close', 'registration', 'first', 'lassafever', 'international', 'conference', '
['merry', 'christmas', 'client', 'partner', 'friend', 'follower', 'new', 'year', 'healthy', 'votehea
['wrapped', 'new', 'year', 'begin', 'reporter', 'take', 'look', 'globalhealth', 'story', 'eager', '
['week', 'twitter', 'like', 'retweet', 'retweet', 'reach', 'new', 'follower', 'see', 'biggest', 'fa
['biggest', 'fan', 'week', 'thank', 'wish', 'family', 'beautiful', 'holiday', 'season', 'merry', 'c
['see', 'kick', 'start', 'next', 'year', 'cordially', 'invite', 'healthforall', 'familyplanning', '
```

*Figure 3-4 Data after pre-processing*

### 3.2.3 Dictionary Generation

After pre-processing our data, we next need to represent our data in a form that can be fed into our topic modelling algorithms. To help us do this, we first generate all unique words in our corpus and assign each word a unique token as shown in Fig. 3-5.



```
{'aaco': 0, 'abajue': 1, 'abandon': 2, 'abanida': 3, 'abaye': 4, 'abc': 5, 'abdallah': 6,
```

*Figure 3-5 Generated Dictionary*

Our algorithms interact with this unique token generated as a representation of the words in the corpus. We used the dictionary module in Python Gensim's library to generate the dictionary of our corpus.

### 3.2.4 Bag-of-Words Generation

Next, we represent our corpus in terms of each document and the unique words in the corpus. Each document takes a unique index also. At the end, a matrix is generated in which column represents a term while each row represents a document and the entries of the matrix represents the frequencies of the terms in the documents. These entries could be raw counts, TF or TF-IDF. To get the bag-of-words representation of our corpus, we used the doc2bow () function of the dictionary module in Python Gensim's library.

### 3.2.5 Topic Modelling

For the mining of our data, we have chosen to use topic modelling. This is because topic models are unsupervised algorithms which are suitable for unlabelled data as ewe have. Secondly, topic models give us insights into corpus, letting us know what our corpus is about in more specific ways beyond a general term like health or sports.

As earlier said, Topic modelling is the process of applying statistical models (topic models) to extract the hidden topics in the data. However, there are several topic modelling techniques available and in other to guide us we have chosen to compare two which have recurrently been recommended as being good for text mining from literature which is Latent Semantic Analysis and Latent Dirichlet Allocation.

### 3.2.6 Latent Semantic Analysis (LSA)

Latent semantic analysis is a technique in natural language processing used to analyse the relationship between members of a corpus and the contained terms (words). LSI assumes closely related terms will appear in similar documents; this is called distributional hypothesis. After generation the document – term matrix from the corpus, a mathematical technique called Singular Value Decomposition (SVD) which is also a dimensionality reduction technique is applied to the document – term matrix which is usually large and sparse to identify patterns and learn the correlation of terms and topics in the corpus. SVD decomposed the document -term matrix into a dot product of three matrices: $A = T \cdot S \cdot D_T$

Where:

m = number of unique terms

n = number of documents

r = Rank of A

A = Document – Term Matrix

T = Term – Topic Matrix

S = Singular Value Matrix

D = Topic – Document matrix

### 3.2.7 Latent Dirichlet Allocation (LDA)

Latent Dirichlet allocation (LDA) is a generative probabilistic model of a corpus (Blei et al., 2003). This means that LDA describes how the corpus is generated using a probabilistic model which precisely is the Dirichlet distribution. The Dirichlet distribution is a probabilistic distribution over other probability distributions. Dirichlet best describes the generation of a corpus form various topics because naturally all topics will not follow the same probability distribution in the generation of the corpus. With LDA, documents are represented as random mixtures over latent topics and each topic is describes by a distribution over words (Blei et al., 2003).

The generative process assumed by LDA is as follows:

1. Choose N ∼ Poisson(ξ).

2. Choose θ ∼ Dir(α).

3. For each of the N words $w_n$:

a) Choose a topic $z_n \sim$ Multinomial(θ).

b) Choose a word $w_n$ from $p(w_n \,|z_n, \beta)$ a multinomial probability conditioned on the topic $z_n$.

Hence, Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z, and a set of N words w is given by:

$$p(\theta, z, w|\alpha, \beta) \; = \; p(\theta|\alpha)\prod_{n=1}^{N} p(z_n|\theta)p(w_n|z_n, \beta)$$

We performed an initial experiment with Nigerian health tweets from 2018 using both LDA and LSA in other to compare performance and make a more informed choice. In guiding our choice of number of topics, we studied the coherence measure of topics over a range 2 and 20 topics as shown in Fig. 3-6 and chose the number of topics with the highest coherence score. Hence, we chose 6 topics for the experiment.

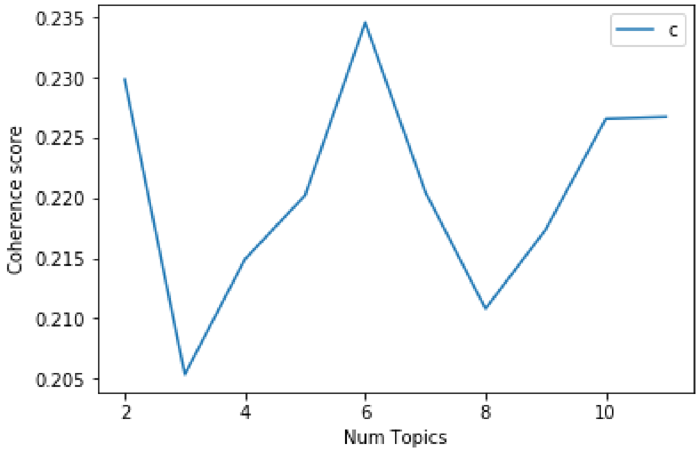
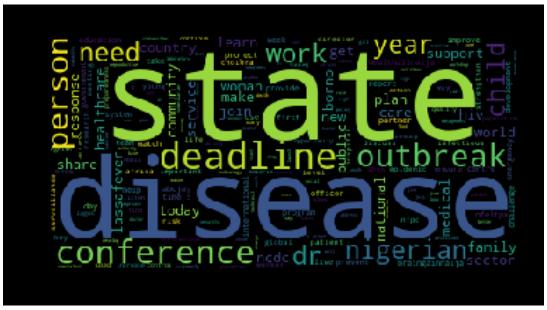


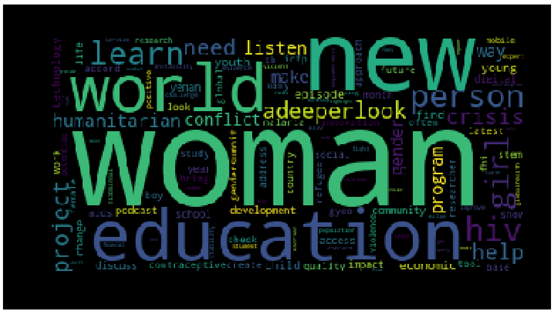
*Figure 3-6 Coherence score over a range of topic numbers*

The output of LSA on the data yielded the following topics displayed as word clouds:

LSI TOPIC #1

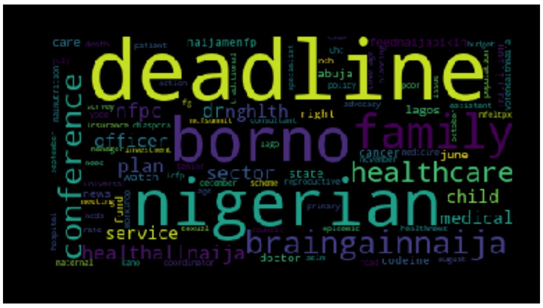
LSI TOPIC #2

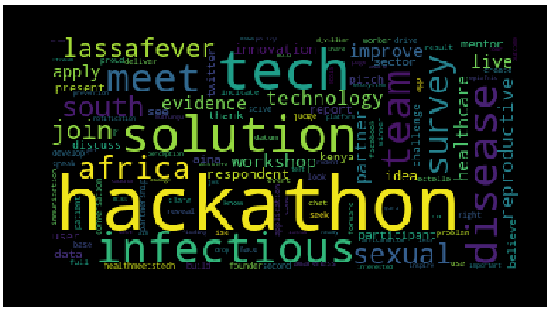




LSI TOPIC #3

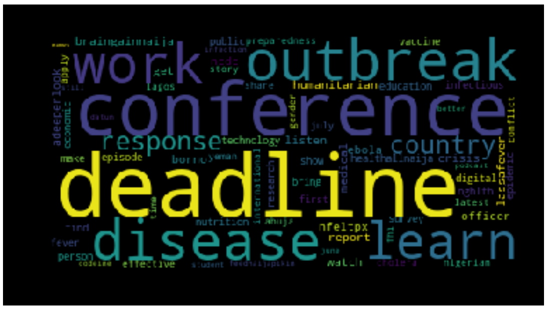
LSI TOPIC #4

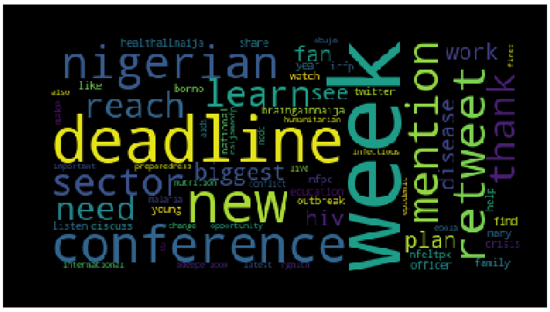




LSI TOPIC #5

LSI TOPIC #6

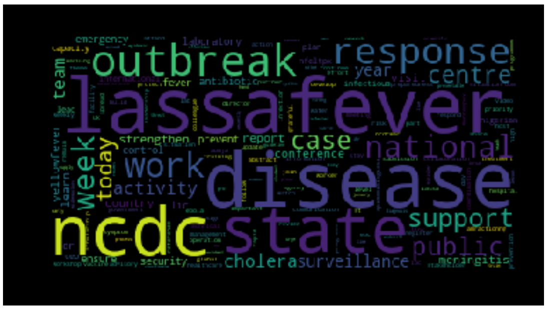


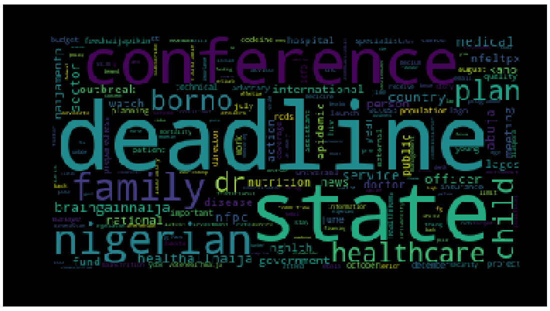


The output of LDA on the data yielded the following topics displayed as word clouds:

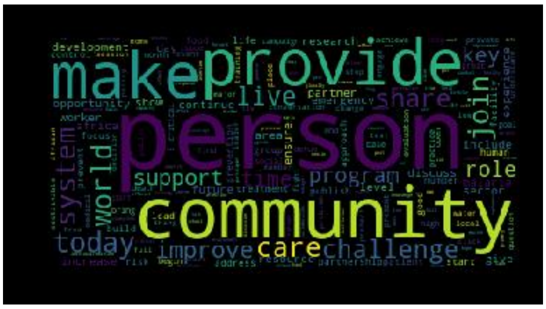
LDA TOPIC #1

LDA TOPIC #2

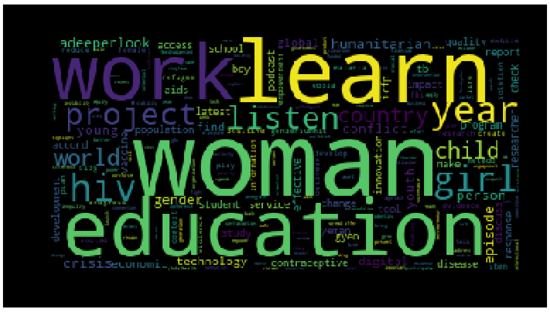




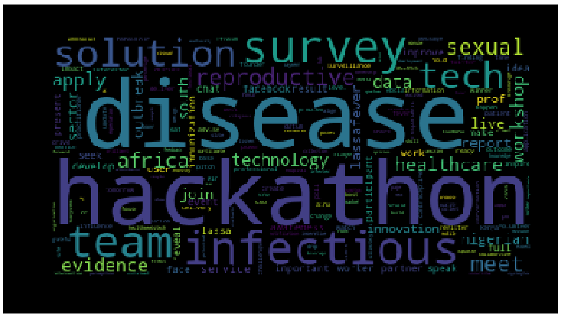
LDA TOPIC #3



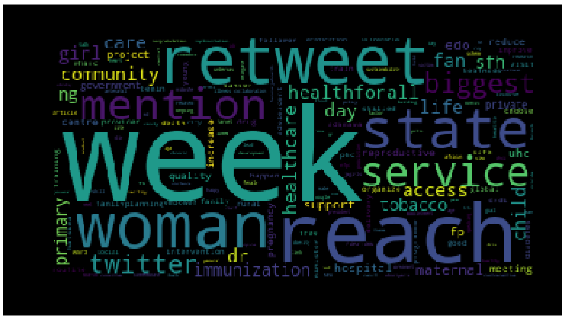
LDA TOPIC #4



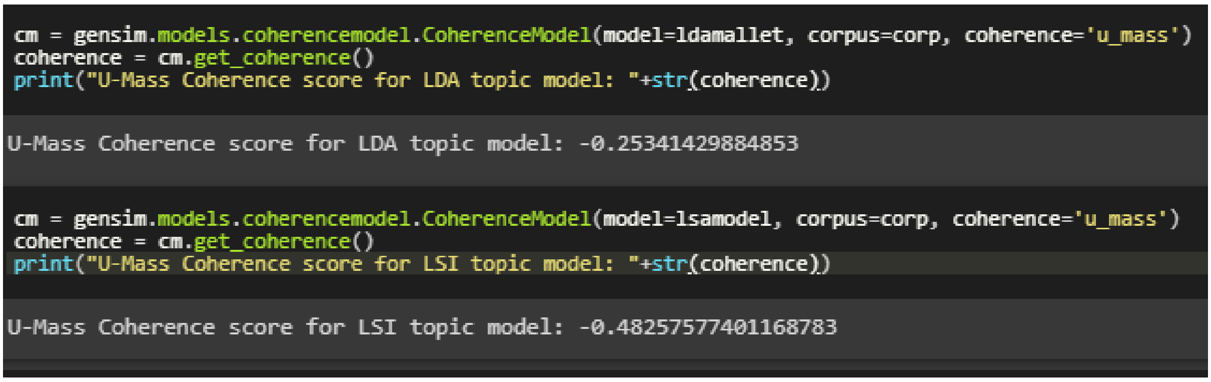
LDA TOPIC #5



LDA TOPIC #6



After our initial experiment, we measured the coherence of both models with U-mass and C_V as shown in Fig. 3-7 and Fig. 3-8 respectively. $U - Mass(-14 < x < 14)$ measures to compare a word only to the preceding and succeeding words respectively while $C\_V(0 < x < 1)$ measures how often the topic words appear together in the corpus. LSI had a U_Mass score of -0.48 and a C_V score of 0.36 while LDA had a U_Mass score of -0.25 and a C_V score of 0.35. This means that LDA yields more coherent topics than LSI.



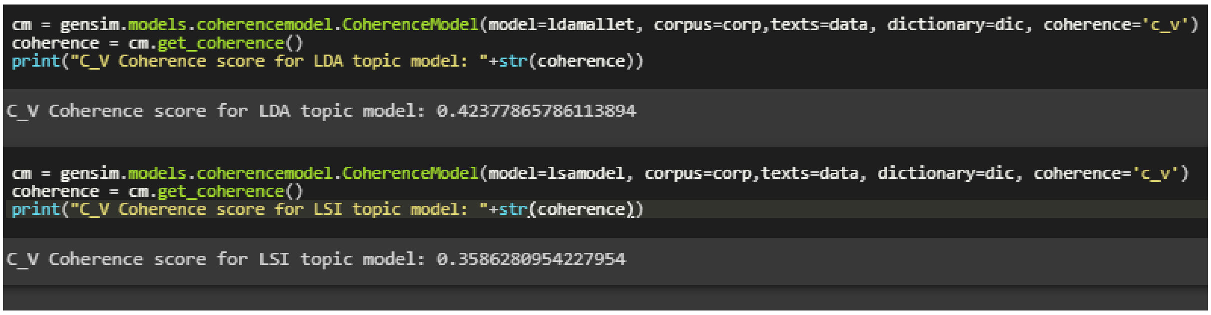
```
cm = gensim.models.coherencemodel.CoherenceModel(model=ldamallet, corpus=corp, coherence='u_mass')
coherence = cm.get_coherence()
print("U-Mass Coherence score for LDA topic model: "+str(coherence))
```

```
U-Mass Coherence score for LDA topic model: -0.25341429884853
```

```
cm = gensim.models.coherencemodel.CoherenceModel(model=lsamodel, corpus=corp, coherence='u_mass')
coherence = cm.get_coherence()
print("U-Mass Coherence score for LSI topic model: "+str(coherence))
```

```
U-Mass Coherence score for LSI topic model: -0.48257577401168783
```

*Figure 3-7 U-Mass Coherence Scores*



```
cm = gensim.models.coherencemodel.CoherenceModel(model=ldamallet, corpus=corp,texts=data, dictionary=dic, coherence='c_v')
coherence = cm.get_coherence()
print("C_V Coherence score for LDA topic model: "+str(coherence))
```

```
C_V Coherence score for LDA topic model: 0.42377865786113894
```

```
cm = gensim.models.coherencemodel.CoherenceModel(model=lsamodel, corpus=corp,texts=data, dictionary=dic, coherence='c_v')
coherence = cm.get_coherence()
print("C_V Coherence score for LSI topic model: "+str(coherence))
```

```
C_V Coherence score for LSI topic model: 0.3586280954227954
```

*Figure 3-8 C_V Coherence Scores*

At the end of this initial experiment and based on the results, we chose LDA for the rest of our experiments.

# 4 RESULT AND DISCUSSION

## 4.1 Introduction

We discuss the results of applying LDA to tweets retrieved for the years 2015, 2016, 2017, 2018 and 2019 in this chapter. Based on the top words that characterize each topic, we attempt to label each topic and then discuss thesis results.

## 4.2 Results

The results for each of our experiments for each year are shown in the below.

### 4.2.1 Results for 2015

We used the following handles for retrieving Nigerian health tweets for 2015 from tweeter: @nighealthwatch, @nmanigeria, @Fmohnigeria, @Nphcdanigeria, @APINNigeria, @EpiAFRIC, @W4HNigeria, @SFHNigeria. To determine the optimal number of topics, we ran LDA on the dataset over a range of 2 to 20 topics and chose the number of topics with the highest coherence which was 4 as shown in Fig. 4-1.

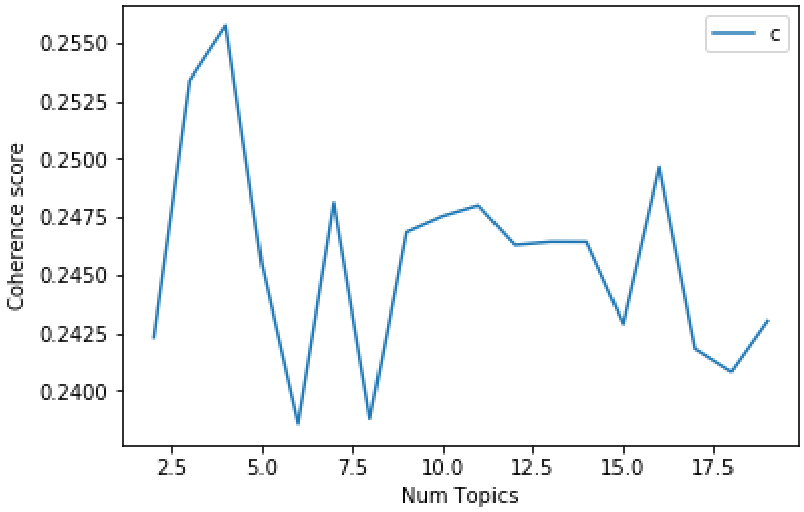
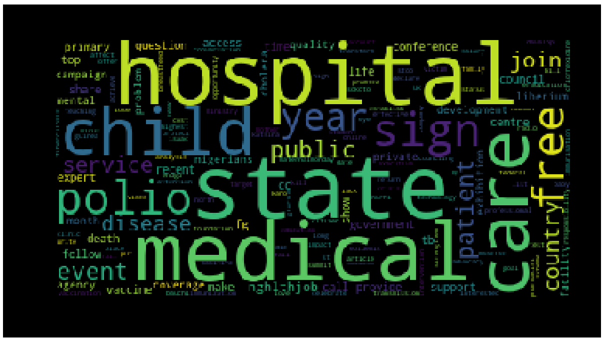


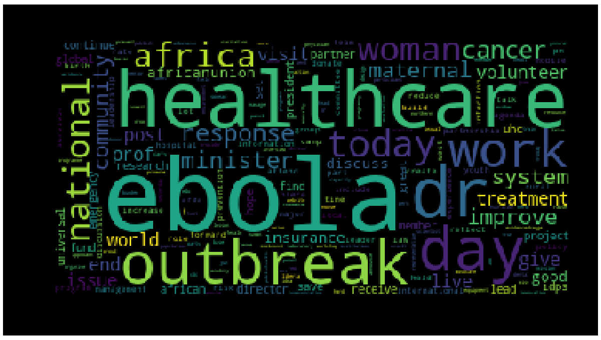
*Figure 4-1 Coherence score for various number of topics 2015*

Choosing 4 topics and running LDA on our data of Nigerian health tweets for 2015 yielded the following results:

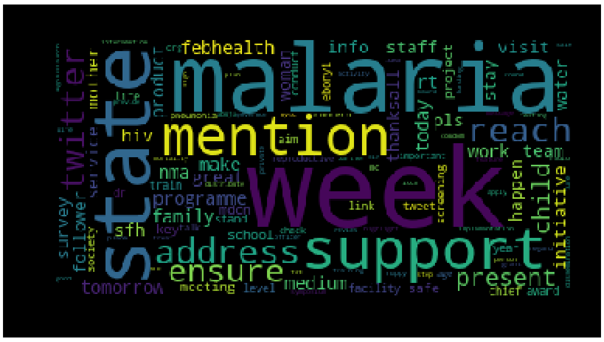
Topic #1



Topic #2



Topic #3



Topic #4

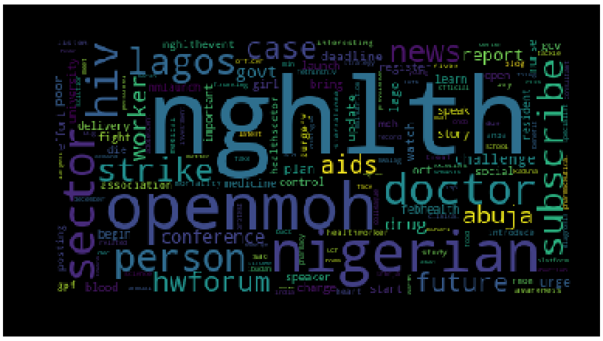


*Table 4.1 Topic Table for Year 2015 Data*

| Topic Number | Top words | Topic Label |
|---|---|---|
| **1** | "state", "hospital, "medical", "care", "child", "polio", "sign", "free", "year", "event" | Polio Campaign |
| **2** | "ebola", "healthcare", "dr", "day", "outbreak", "work", "today", "woman", "national", "africa" | Ebola Outbreak |
| **3** | "week", "malaria", "state", "support", "mention", "address", "reach", "twitter", "ensure", "present" | Malaria campaign |
| **4** | "nghlth", "openmoh", "nigerian", "sector", "hiv", "doctor", "person", "lagos", "subscribe", "strike" | Medical Doctor strike |

### 4.2.2 Results for 2016

We used the following handles for retrieving Nigerian health tweets for 2015 from tweeter: @nighealthwatch, @nmanigeria, @Fmohnigeria, @NphcdaNG, @EpiAFRIC, @APINNigeria, @W4HNigeria, @SFHNigeria, @NCDCgov.

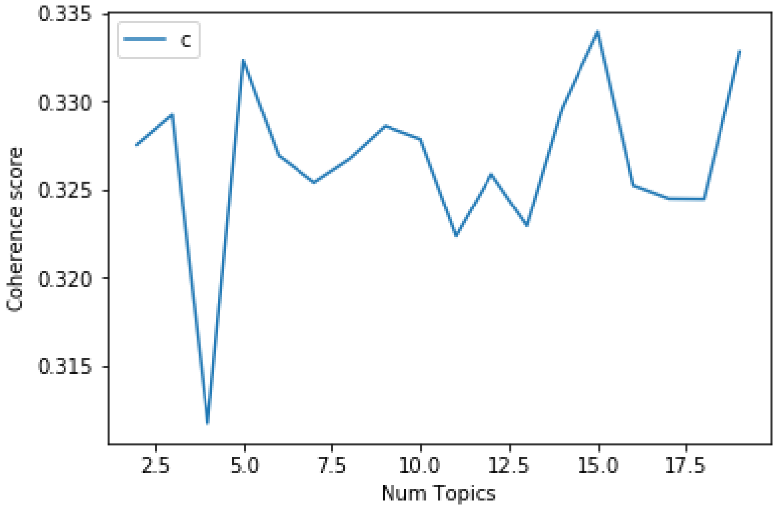
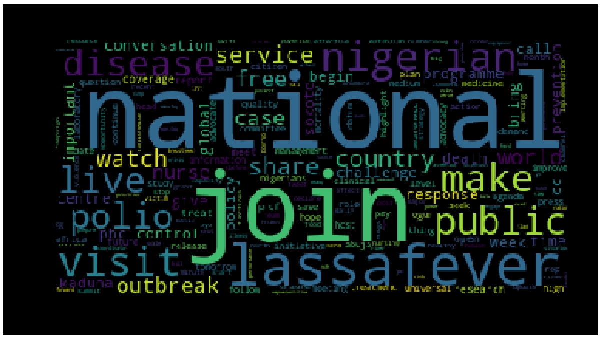


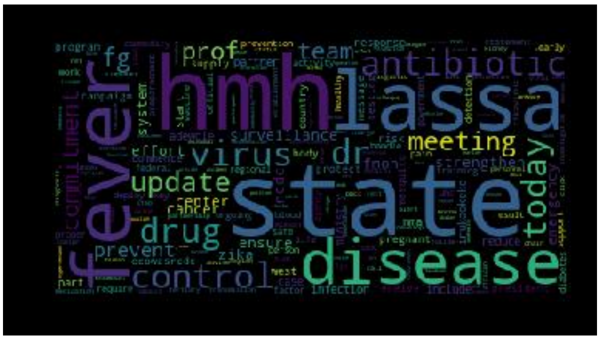
*Figure 4-2 Coherence Scores for various number of topics 2016*

Choosing 5 topics based on the output shown in Fig. 4-2 and running LDA on our data of Nigerian health tweets for 2016 yielded the following results.

Topic #1

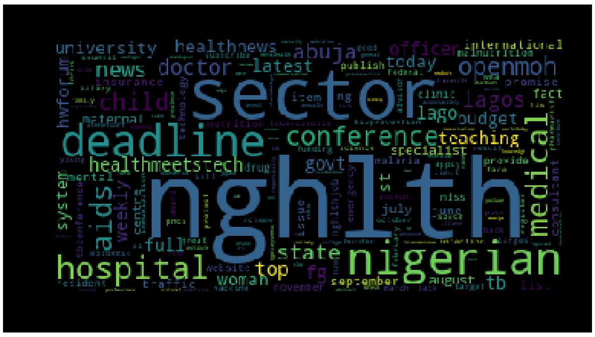


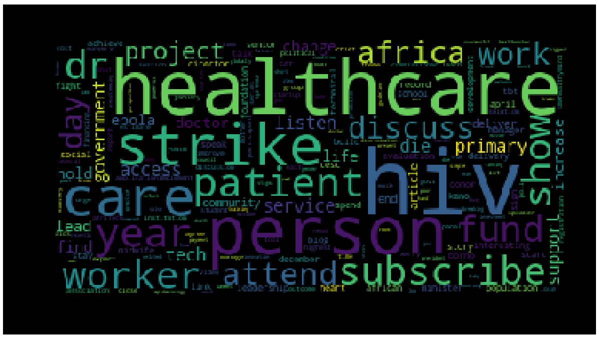
Topic #2



Topic #3

Topic #4





Topic #5

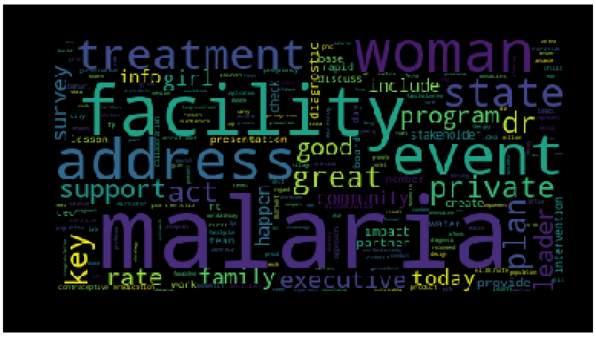


*Table 4.1 Topic Table for Year 2016*

| Topic Number | Top words | Topic Label |
|---|---|---|
| **1** | "national", "join", "lassafever", "nigerian", "visit", "make", "public", "live", "disease", "polio" | Polio Free |
| **2** | "hmh", "state", "lassa", "fever", "disease", "virus" "antibiotic", "dr", "control", "drug" | Lassa fever outbreak |
| **3** | "nghlth", "sector", "nigerian", "deadline", "hospital", "medical", "aids", "conference", "openmoh", "child" | Medical conference |
| **4** | "hiv", "healthcare", "person", "strike", "care", "patient", "year", "subscribe", "worker", "dr" | Health workers strike |
| **5** | "malaria", "facility", "woman", "address", "event", "treatment", "state", "private", "great", "plan" | Malaria treatment / campaign |

### 4.2.3 Results for 2017

We used the following handles for retrieving Nigerian health tweets for 2017 from tweeter: @nighealthwatch, @nmanigeria, @Fmohnigeria, @NphcdaNG, @EpiAFRIC, @APINNigeria, @W4HNigeria, @SFHNigeria, @NCDCgov, @wharc_nigeria and @WHONigeria.

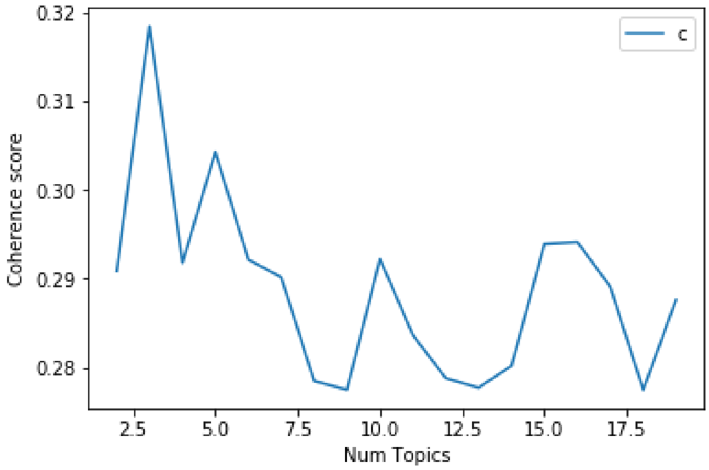
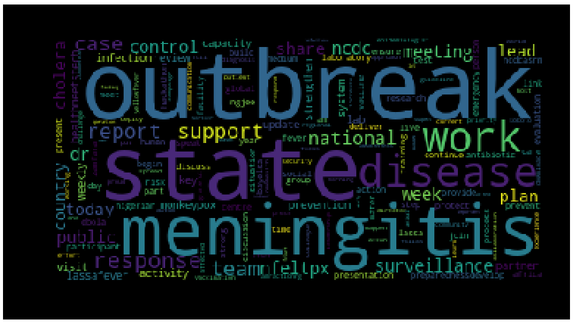


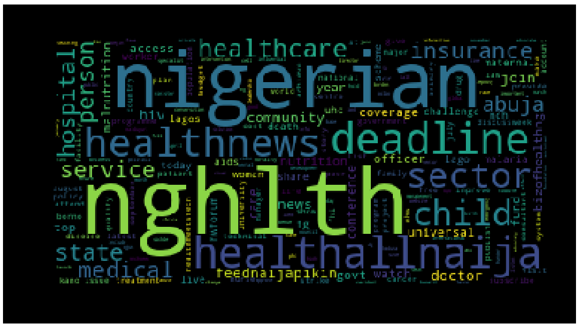
*Figure 4.3 Coherence score for various number of topics 2017*

Choosing 3 topics based on the output shown in Fig. 4-3 and running LDA on our data of Nigerian health tweets for 2017 yielded the following results:

Topic #1

Topic #2





Topic #3

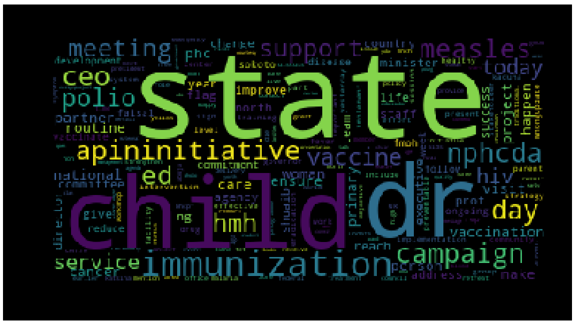


*Table 4.2 Topic Table for Year 2017*

| Topic Number | Top Words | Topic Label |
|---|---|---|
| **1** | "outbreak", "state", "meningitis", "disease", "work", "response", "team", "support", "case", "report" | Meningitis Outbreak |
| **2** | "nghlth", "nigerian", "healthallnaija", "deadline", "healthnews", "sector", "child", "healthcare", "state", "hospital" | Medical Insurance |
| **3** | "state", "dr", "child", "immunization", "apininitiative", "ceo", "campaign", "nphcda", "ed", "day" | Child Immunization |

### 4.2.4 Results for 2018

We used the following handles for retrieving Nigerian health tweets for 2018 from tweeter: @nighealthwatch, @nmanigeria, @Fmohnigeria, @NphcdaNG, @EpiAFRIC, @APINNigeria, @W4HNigeria, @SFHNigeria, @NCDCgov, @wharc_nigeria and @WHONigeria.

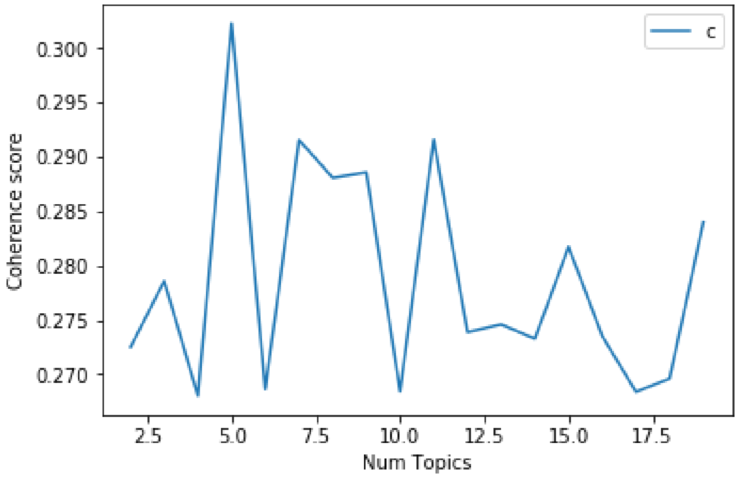
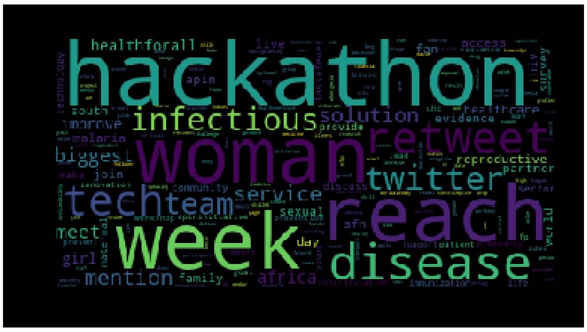


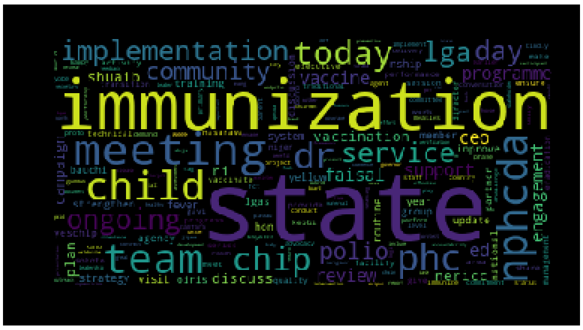
*Figure 3.4 Coherence Score for various number of Topics 2018*

Choosing 5 topics based on the output shown in Fig. 4-4 and running LDA on our data of Nigerian health tweets for 2018 yielded the following results.

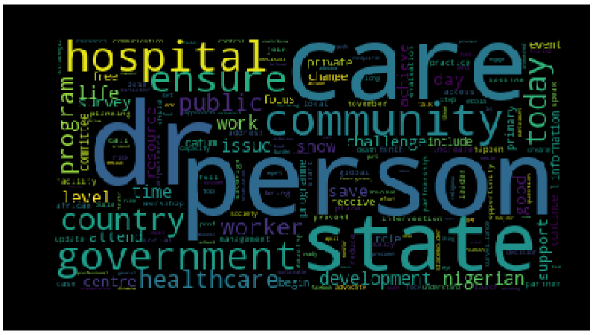
Topic #1



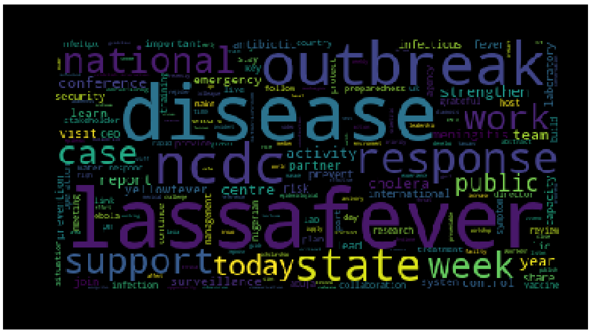
Topic #2



Topic #3



Topic #4



Topic #5

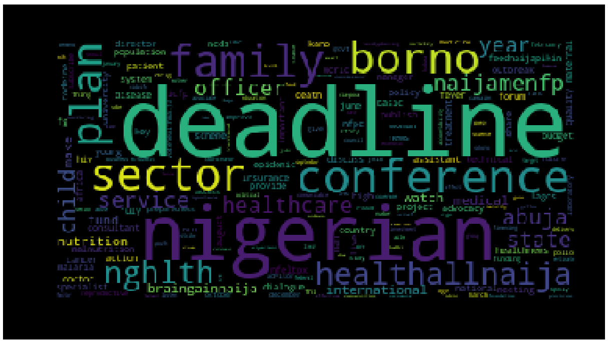


*Table 4.4 Topic Table for Year 2018*

| Topic Number | Top Words | Topic Label |
|---|---|---|
| 1 | "hackathon", "woman", "week", "reach", "disease", "retweet", "tech", "twitter", "infectious", "team" | E – Health solution |
| 2 | "state", "immunization", "meeting", "team", "nphcda", "child", "dr", "phc", "chip", "today" | Child Immunization |
| 3 | "dr", "person", "care", "state", "community", "hospital", "government", "ensure", "country", "today" | Community healthcare |
| 4 | "lassafever", "disease", "outbreak", "ncdc", "response", "state", "support", "national", "week", "work" | Lassa fever outbreak |
| 5 | "deadline", "nigerian", "conference", "sector", "family", "plan", "borno", "healthallnaija", "nghlth", "naijamenfp" | Family planning |

### 4.2.5 Results for 2019

We used the following handles for retrieving Nigerian health tweets for 2019 from tweeter: @nighealthwatch, @nmanigeria, @Fmohnigeria, @NphcdaNG, @EpiAFRIC, @APINNigeria, @W4HNigeria, @SFHNigeria, @NCDCgov, @wharc_nigeria and @WHONigeria.

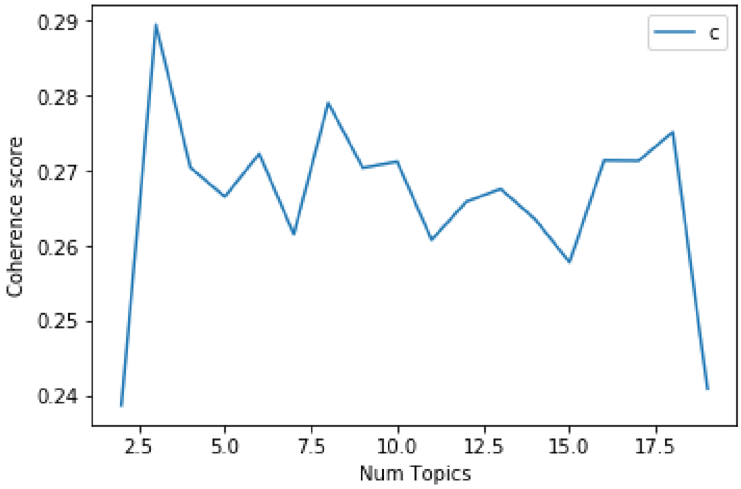
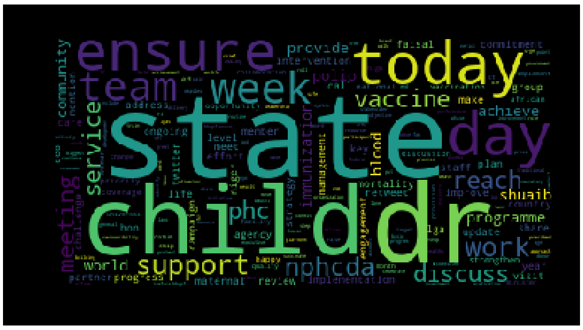


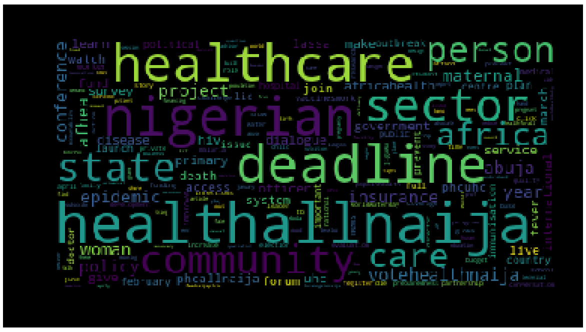
*Figure 4.5 Coherence score for various number of topics 2019*

Choosing 3 topics based on the output shown in Fig. 4-5 and running LDA on our data of Nigerian health tweets for 2019 yielded the following results.

Topic #1



Topic #2



Topic #3

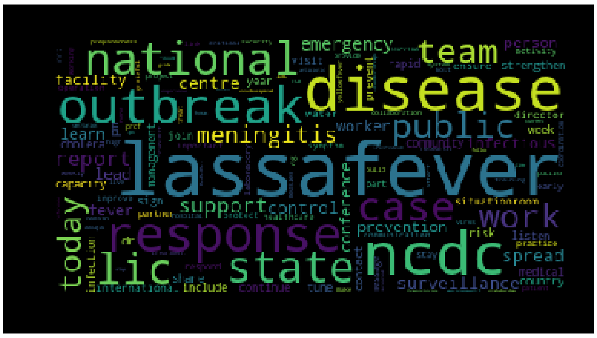


*Table 4.4 Topic Table for Year 2019*

| Topic Number | Top Words | Topic Labels |
|---|---|---|
| **1** | "state", "dr", "child", "today", "day", "ensure", "week", "team", "support", "work" | Child vaccination |

| | | |
|---|---|---|
| 2 | "healthallnaija", "nigerian", "deadline", "healthcare", "sector", "community", "state", "person", "care”, "africa" | Community healthcare |
| 3 | "lassafever”, "disease", "ncdc", "outbreak", "response", "national", "state", "lic", "case", "public" | Lassa fever outbreak |

## 4.3 Discussion

The results of our experiments show case some trends in the Nigerian health sector over the past 5 years. There have been repeated cases of Lassa fever outbreak in years 2016, 2018 and 2019 which reports by the World Health Organization (WHO) (WHO, 2017, 2018, 2019) confirms. The results also show that there has been a trend on children immunization / vaccination and community healthcare.

We also attempted the label the outputted topics in Table 4-1, Table 4-2, Table 4-3, Table 4-4 and Table 4-5. This is only an attempt using our intuition and may not be entire accurate since the labelling was not done by a domain expert.

# 5 CONCLUSION AND RECOMMENDATION

## 5.1 CONCLUSION

In these times, availability of data is not an issue. Utilization of the available data however is still very low. We can draw insights on different subjects from Twitter data. We demonstrated this by applying LDA topic model on Nigerian health tweets from 2015 to 2019.

From our experiments, we detected the outbreaks of Ebola, Meningitis and Lassa fever within this time frame. We also observed the reoccurring topic of child immunization/vaccination over the years.

Individuals, governments and organizations have a great asset in their Twitter data and should utilize it through text mining techniques like LDA topic model.

## 5.2 RECOMMENDATION

Getting the topics contained in a Twitter dataset could be used as a tool for labelling Twitter data. Hence, I recommend using the topics derived from topic modelling to label the Twitter dataset and performing some supervised learning techniques. With labelled dataset, more algorithms can be explored, and performance measured.

## References

Aggarwal, C. C. (2015). Data Mining. In *Springer; 2015 edition* (1st ed.). https://doi.org/10.1007/978-3-319-14142-8

Ahuja, A., Wei, W., & Carley, K. M. (2016). Microblog Sentiment Topic Model. *IEEE International Conference on Data Mining Workshops, ICDMW*, 1031–1038. https://doi.org/10.1109/ICDMW.2016.0149

Andrius Velykis. (2018). tint - Tokenization and sentence splitting. Retrieved April 9, 2019, from http://tint.fbk.eu/tokenization.html

Anjaria, M., & Guddeti, R. M. R. (2014). Influence factor based opinion mining of Twitter data using supervised learning. *2014 6th International Conference on Communication Systems and Networks, COMSNETS 2014*. https://doi.org/10.1109/COMSNETS.2014.6734907

Asghari, M., Sierra-Sosa, D., & Elmaghraby, A. (2019). Trends on Health in Social Media: Analysis using Twitter Topic Modeling. *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, (December), 558–563. https://doi.org/10.1109/isspit.2018.8642679

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993

Boyd-Graber, Jordan & Hu, Yuening & Mimno, D. (2017). Applications of Topic Models. *Foundations and Trends® in Information Retrieval*, *11*(1–2), 143–296. https://doi.org/10.1561/1500000030

Dumais, S. T. (2005). Latent semantic analysis. *Annual Review of Information Science and Technology*, *38*(1), 188–230. https://doi.org/10.1002/aris.1440380105

Halibas, A. S., Shaffi, A. S., & Mohamed, M. A. K. V. (2018). Application of text classification and clustering of Twitter data for business analytics. *Proceedings of Majan International Conference: Promoting Entrepreneurship and Technological Skills: National Needs, Global Trends, MIC 2018*, 1–7. https://doi.org/10.1109/MINTC.2018.8363162

Hassan, D. (2018). A text mining approach for evaluating event credibility on twitter. *Proceedings - 2018 IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2018*, 175–178. https://doi.org/10.1109/WETICE.2018.00039

He, L., Jia, Y., Han, W., & Ding, Z. (2014). Mining user interest in microblogs with a user-topic model. *China Communications*, *11*(8), 131–144. https://doi.org/10.1109/CC.2014.6911095

Hidayatullah, A. F., Pembrani, E. C., Kurniawan, W., Akbar, G., & Pranata, R. (2018). Twitter Topic Modeling on Football News. *2018 3rd International Conference on Computer and Communication Systems, ICCCS 2018*, 94–98. https://doi.org/10.1109/CCOMS.2018.8463231

Lansley, G., & Longley, P. A. (2016). The geography of Twitter topics in London. *Elsevier*, 1–12.

Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013). *Improving LDA topic models for microblogs via tweet pooling and automatic labeling*. 889. https://doi.org/10.1145/2484028.2484166

Nirmala, C. R., Roopa, G. M., & Kumar, K. R. N. (2015). Twitter data analysis for unemployment crisis. *Proceedings of the 2015 International Conference on Applied and Theoretical Computing and Communication Technology, ICATccT 2015*, 420–

423. https://doi.org/10.1109/ICATCCT.2015.7456920

Portland Communications. (2016). How Africa Tweets 2016. Retrieved April 9, 2019, from https://portland-communications.com/publications/how-africa-tweets-2016/

Provost, F., & Fawcett, T. (2016). *Data Science for Business*. O'Reilly Media Inc.

Ramadhani, A. M., & Goo, H. S. (2017). Twitter sentiment analysis using deep learning methods. *Proceedings - 2017 7th International Annual Engineering Seminar, InAES 2017*. https://doi.org/10.1109/INAES.2017.8068556

Ramanathan, V., & Meyyappan, T. (2019). Twitter text mining for sentiment analysis on people's feedback about Oman tourism. *2019 4th MEC International Conference on Big Data and Smart City, ICBDSC 2019*, 1–5. https://doi.org/10.1109/ICBDSC.2019.8645596

Samarawickrama, S., Karunasekera, S., & Harwood, A. (2015). Finding high-level topics and tweet labeling using topic models. *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, 242–249. https://doi.org/10.1109/ICPADS.2015.38

Sapul, M. S. C., Aung, T. H., & Jiamthapthaksin, R. (2017). Trending topic discovery of Twitter Tweets using clustering and topic modeling algorithms. *Proceedings of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017*. https://doi.org/10.1109/JCSSE.2017.8025911

Terragon Group. (2018). A look at Terragon Group's 2018 Digital Trends for Nigeria. Retrieved April 9, 2019, from https://nairametrics.com/2018/07/17/report-instagram-dominates-as-22-million-nigerians-use-internet-for-social-media/

The Stanford Natural Language Processing Group. (n.d.). Retrieved April 14, 2019, from https://nlp.stanford.edu/software/CRF-NER.html

Tripathi, P., Vishwakarma, S. K., & Lala, A. (2015). Sentiment Analysis of English Tweets Using Rapid Miner. *Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015*, 668–672. https://doi.org/10.1109/CICN.2015.137

Twitter Usage Statistics - Internet Live Stats. (n.d.). Retrieved May 17, 2019, from https://www.internetlivestats.com/twitter-statistics/

WHO. (2017). *Emergencies preparedness , response Lassa Fever – Nigeria*. (January), 2017–2019. Retrieved from http://who.int/csr/don/27-january-2016-lassa-fever-nigeria/en/

WHO. (2018). *Emergencies preparedness , response Lassa Fever – Nigeria*. (January), 2018–2020. Retrieved from http://who.int/csr/don/27-january-2016-lassa-fever-nigeria/en/

WHO. (2019). *Emergencies preparedness , response Lassa Fever – Nigeria*. (January), 1–3. Retrieved from http://who.int/csr/don/27-january-2016-lassa-fever-nigeria/en/

Winans, M., Faupel, D., Armstrong, A., Henderson, J., Valentine, E., McDonald, L., … Magill, E. (2017). 10 Key Marketing Trends for 2017 Customer Expectations and Ideas for Exceeding Customer Expectations. *Ibm*, 1–18. Retrieved from https://public.dhe.ibm.com/common/ssi/ecm/wr/en/wrl12345usen/watson-customer-engagement-watson-marketing-wr-other-papers-and-reports-wrl12345usen-20170719.pdf

Wold, H. M., Vikre, L., Gulla, J. A., Özgöbek, Ö., & Su, X. (2016). *Twitter Topic Modeling for Breaking News Detection*. *2*(Webist), 211–218. https://doi.org/10.5220/0005801902110218

Wolny, W. (2016). Knowledge Gained from Twitter Data. *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems*, *8*, 1133–

1136. https://doi.org/10.15439/2016f149

Xu, J. (2018). Topic Modeling with LSA, PSLA, LDA &amp; lda2Vec – NanoNets – Medium. Retrieved April 22, 2019, from Medium website: https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05

Yang, S., & Zhang, H. (2018). Text Mining of Twitter Data Using a Latent Dirichlet Allocation Topic Model and Sentiment Analysis. *International Journal of Computer and Information Engineering*, *12*(7), 525–529. https://doi.org/10.5281/zenodo.1317350

Zhao, W. X., Jiang, J., Weng, J., He, J., Lim, E.-P., Yan, H., & Li, X. (2011). *Comparing Twitter and Traditional Media Using Topic Models*. 338–349. https://doi.org/10.1007/978-3-642-20161-5_34

Zulfikar, W. B., Irfan, M., Alam, C. N., & Indra, M. (2017). The comparation of text mining with Naive Bayes classifier, nearest neighbor, and decision tree to detect Indonesian swear words on Twitter. *2017 5th International Conference on Cyber and IT Service Management, CITSM 2017*. https://doi.org/10.1109/CITSM.2017.8089231

## Appendix

```
# -*- coding: utf-8 -*-
"""Untitled1.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1FXsH21aNIRvn_UN3dpEjpPupzt82pCXV
"""

!wget "https://www.machinelearningplus.com/wp-content/uploads/2018/03/mallet-
2.0.8.zip"
!unzip "mallet-2.0.8.zip"
!pip install wordcloud
from wordcloud import WordCloud, STOPWORDS

#Install pyLDAvis to interpret the topics in a topic model
!pip install pyldavis
!pip install GetOldTweets3
!python3 -m spacy download en
!pip install spacy
!sudo apt-get install libmysqlclient-dev
!pip install Pattern

#Import base packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
#Libraries for data retrieval and cleaning
import GetOldTweets3 as got
import string
import re
#Libraries for data pre-processing and topic modelling
import spacy
spacy.load('en')
import gensim
import nltk
nltk.download('punkt')
from nltk.corpus import wordnet
from nltk import word_tokenize
from nltk.corpus import stopwords
nltk.download('stopwords')
#Libraries for visualization
import pyLDAvis.gensim
pyLDAvis.enable_notebook()
# Ignore warnings
import warnings
warnings.filterwarnings('ignore')
#To enable reading in files to colab
# from google.colab import files
# uploaded = files.upload()
# import io
#"fhi360","wharc_nigeria","WHONigeria"

#functions to change tweets to list of lists each representing a tweet
def atweet2lst(tweet):
  return [tweet.id, tweet.permalink, tweet.username, tweet.to, tweet.text,
tweet.date, tweet.retweets, tweet.favorites, tweet.mentions, tweet.hashtags,
tweet.geo]

def allhandletweets2lst(tweets):
  container = []
  for tweet in tweets:
      container.append(atweet2lst(tweet))
  return container

#Official twitter hanldes
handles=["nighealthwatch", "nmanigeria", "Fmohnigeria", "NphcdaNG", "EpiAFRIC",
"APINNigeria", "W4HNigeria", "SFHNigeria"]
#Information derived from each tweet
cols = ["ident", "permalink", "username", "to", "text", "date", "retweets",
"favorites", "mentions"," hashtags", "geo"]

#save all tweets from a handle as a CSV file
for i in range(len(handles)):
    tweetCriteria =
got.manager.TweetCriteria().setUsername(handles[i]).setSince("2015-01-
01").setUntil("2015-12-31")
    tweets = got.manager.TweetManager.getTweets(tweetCriteria)
    pd.DataFrame(allhandletweets2lst(tweets), columns=cols).to_csv(
handles[i]+"2015.csv", index=False)

#Official twitter hanldes
handles=["nighealthwatch", "nmanigeria", "Fmohnigeria", "NphcdaNG", "EpiAFRIC",
"APINNigeria", "W4HNigeria", "SFHNigeria", "NCDCgov"]
#Information derived from each tweet
cols = ["ident", "permalink", "username", "to", "text", "date", "retweets",
"favorites", "mentions"," hashtags", "geo"]

#save all tweets from a handle as a CSV file
for i in range(len(handles)):
    tweetCriteria =
got.manager.TweetCriteria().setUsername(handles[i]).setSince("2016-01-
01").setUntil("2016-12-31")
    tweets = got.manager.TweetManager.getTweets(tweetCriteria)
    pd.DataFrame(allhandletweets2lst(tweets), columns=cols).to_csv(
handles[i]+"2016.csv", index=False)

#Official twitter hanldes
handles=["nighealthwatch", "nmanigeria", "Fmohnigeria", "NphcdaNG", "EpiAFRIC",
"APINNigeria", "W4HNigeria", "SFHNigeria", "NCDCgov","wharc_nigeria","WHONigeria"]
#Information derived from each tweet
cols = ["ident", "permalink", "username", "to", "text", "date", "retweets",
"favorites", "mentions"," hashtags", "geo"]

#save all tweets from a handle as a CSV file
for i in range(len(handles)):
    tweetCriteria =
got.manager.TweetCriteria().setUsername(handles[i]).setSince("2017-01-
01").setUntil("2017-12-31")
    tweets = got.manager.TweetManager.getTweets(tweetCriteria)
    pd.DataFrame(allhandletweets2lst(tweets), columns=cols).to_csv(
handles[i]+"2017.csv", index=False)
```

```
#Official twitter hanldes
handles=["nighealthwatch", "nmanigeria", "Fmohnigeria", "NphcdaNG", "EpiAFRIC",
"APINNigeria", "W4HNigeria", "SFHNigeria", "NCDCgov","wharc_nigeria","WHONigeria"]
#Information derived from each tweet
cols = ["ident", "permalink", "username", "to", "text", "date", "retweets",
"favorites", "mentions"," hashtags", "geo"]

#save all tweets from a handle as a CSV file
for i in range(len(handles)):
    tweetCriteria =
got.manager.TweetCriteria().setUsername(handles[i]).setSince("2018-01-
01").setUntil("2018-12-31")
    tweets = got.manager.TweetManager.getTweets(tweetCriteria)
    pd.DataFrame(allhandletweets2lst(tweets), columns=cols).to_csv(
handles[i]+"2018.csv", index=False)

#Official twitter hanldes
handles=["nighealthwatch", "nmanigeria", "Fmohnigeria", "NphcdaNG", "EpiAFRIC",
"APINNigeria", "W4HNigeria", "SFHNigeria", "NCDCgov","wharc_nigeria","WHONigeria"]
#Information derived from each tweet
cols = ["ident", "permalink", "username", "to", "text", "date", "retweets",
"favorites", "mentions"," hashtags", "geo"]

#save all tweets from a handle as a CSV file
for i in range(len(handles)):
    tweetCriteria =
got.manager.TweetCriteria().setUsername(handles[i]).setSince("2019-01-
01").setUntil("2019-12-31")
    tweets = got.manager.TweetManager.getTweets(tweetCriteria)
    pd.DataFrame(allhandletweets2lst(tweets), columns=cols).to_csv(
handles[i]+"2019.csv", index=False)

#Function to remove URL's, stopwords, common words and numbers from tweets.
#And to tokenize and lemmatize
def clean(astring):
  astring.lower()
  astring = re.sub(r'(\d+|https?://\S+|#|@[A-Za-z0-
9]+|twitter.com/\S+|pic|nhw\S+|NHW\S+|job)', "", astring)
  temp = ""
  result = []
  astring = gensim.utils.lemmatize(astring)
  for _ in astring:
    temp += ((re.match("(\w+)", _.decode("utf-8")).group(0)) + " ")
  temp = word_tokenize(temp)
  stopPuncWords = stopwords.words('english') + list(string.punctuation)

stopPuncWords.extend(["health","nigeria","pic","nhwjob","job","vacancy","read","detai
l"])
  result = [i for i in temp if i not in stopPuncWords]
  return result

#Function to convert series into a list
def intoalist(series):
  datalist = []
  for _ in series:
    datalist += _
  return datalist
```

```
def compute_coherence_values(dictionary, corpus, texts, limit, start=2, step=3):
    """
    Compute c_v coherence for various number of topics

    Parameters:
    ----------
    dictionary : Gensim dictionary
    corpus : Gensim corpus
    texts : List of input texts
    limit : Max num of topics

    Returns:
    -------
    model_list : List of LDA topic models
    coherence_values : Coherence values corresponding to the LDA model with
respective number of topics
    """
    coherence_values = []
    model_list = []
    for num_topics in range(start, limit, step):
        model=gensim.models.ldamodel.LdaModel(corpus=doc_term_matrix,
id2word=dictionary, num_topics=num_topics)
        model_list.append(model)
        coherencemodel = gensim.models.CoherenceModel(model=model, texts=texts,
dictionary=dictionary, coherence='c_v')
        coherence_values.append(coherencemodel.get_coherence())

    return model_list, coherence_values

"""# For 2015"""

# Read in CSV files into DataFrames for each twitter handle
nighealthwatch= pd.read_csv("nighealthwatch2015.csv")
nmanigeria= pd.read_csv("nmanigeria2015.csv")
Fmohnigeria= pd.read_csv("Fmohnigeria2015.csv")
Nphcdanigeria= pd.read_csv("NphcdaNG2015.csv")
APINNigeria= pd.read_csv("APINNigeria2015.csv")
EpiAFRIC= pd.read_csv("EpiAFRIC2015.csv")
W4HNigeria= pd.read_csv("W4HNigeria2015.csv")
SFHNigeria= pd.read_csv("SFHNigeria2015.csv")

#Retrieving only tweet text information for each twitter handle
nighealthwatchtweet= nighealthwatch["text"]
nmanigeriatweet= nmanigeria["text"]
Fmohnigeriatweet= Fmohnigeria["text"]
Nphcdanigeriatweet= Nphcdanigeria["text"]
APINNigeriatweet= APINNigeria["text"]
EpiAFRICtweet= EpiAFRIC["text"]
W4HNigeriatweet= W4HNigeria["text"]
SFHNigeriatweet= SFHNigeria["text"]

#Pre-process tweets
newnighealthwatch= intoalist(nighealthwatchtweet.apply(clean))
newnmanigeria= intoalist(nmanigeriatweet.apply(clean))
newFmohnigeria= intoalist(Fmohnigeriatweet.apply(clean))
newNphcdanigeria= intoalist(Nphcdanigeriatweet.apply(clean))
newAPINNigeria= intoalist(APINNigeriatweet.apply(clean))
newEpiAFRIC= intoalist(EpiAFRICtweet.apply(clean))
```

```
newW4HNigeria= intoalist(W4HNigeriatweet.apply(clean))
newSFHNigeria= intoalist(SFHNigeriatweet.apply(clean))

data = [newnighealthwatch, newnmanigeria, newFmohnigeria, newNphcdanigeria,
newAPINNigeria, newEpiAFRIC,  newW4HNigeria, newSFHNigeria]

#Dictionary encapsulates the mapping between
#normalized words and their integer ids.
dictionary = gensim.corpora.Dictionary(data)
dictionary.save('dictionary.dict')
print( dictionary)

#Convert document (a list of words) into the bag-of-words format
#list of (token_id, token_count) 2-tuples.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data]

#The Matrix Market (MM) exchange formats provide a simple mechanism to facilitate the
exchange of matrix data
gensim.corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

model_list, coherence_values = compute_coherence_values(dictionary=dictionary,
corpus=doc_term_matrix, texts=data, start=2, limit=20, step=1)
# Show graph
limit=20; start=2; step=1;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

num_topic = 4
Lda = gensim.models.LdaMulticore#LdaMulticore uses all CPU cores to parallelize and
speed up model training
lda= Lda(doc_term_matrix, num_topics=num_topic,id2word = dictionary,
passes=20,chunksize=100,random_state=3)

lda.save("lda_topic_model")

for i in lda.print_topics():
    for j in i: print(j)

dic = gensim.corpora.Dictionary.load('dictionary.dict')
corp = gensim.corpora.MmCorpus('corpus.mm')
lda_tp = gensim.models.LdaModel.load("lda_topic_model")

viz= pyLDAvis.gensim.prepare(lda_tp, corp, dic)
viz

ldamallet = gensim.models.wrappers.LdaMallet('mallet-2.0.8/bin/mallet', corpus= corp,
num_topics= 4, id2word= dictionary)
# LDA Mallet Model
for i in ldamallet.print_topics():
    for j in i: print(j)

for t in range(ldamallet.num_topics):
    plt.figure()
    plt.imshow(WordCloud().fit_words(dict(ldamallet.show_topic(t, 200))))
    plt.axis("off")
    plt.title("Topic #" + str(t))
    plt.show()

"""# For 2016"""

# Read in CSV files into DataFrames for each twitter handle
nighealthwatch= pd.read_csv("nighealthwatch2016.csv")
nmanigeria= pd.read_csv("nmanigeria2016.csv")
Fmohnigeria= pd.read_csv("Fmohnigeria2016.csv")
Nphcdanigeria= pd.read_csv("NphcdaNG2016.csv")
APINNigeria= pd.read_csv("APINNigeria2016.csv")
EpiAFRIC= pd.read_csv("EpiAFRIC2016.csv")
W4HNigeria= pd.read_csv("W4HNigeria2016.csv")
SFHNigeria= pd.read_csv("SFHNigeria2016.csv")
NCDCgov= pd.read_csv("NCDCgov2016.csv")

#Retrieving only tweet text information for each twitter handle
nighealthwatchtweet= nighealthwatch["text"]
nmanigeriatweet= nmanigeria["text"]
Fmohnigeriatweet= Fmohnigeria["text"]
Nphcdanigeriatweet= Nphcdanigeria["text"]
APINNigeriatweet= APINNigeria["text"]
EpiAFRICtweet= EpiAFRIC["text"]
W4HNigeriatweet= W4HNigeria["text"]
SFHNigeriatweet= SFHNigeria["text"]
NCDCgovtweet= NCDCgov["text"]

#Pre-process tweets
newnighealthwatch= intoalist(nighealthwatchtweet.apply(clean))
newnmanigeria= intoalist(nmanigeriatweet.apply(clean))
newFmohnigeria= intoalist(Fmohnigeriatweet.apply(clean))
newNphcdanigeria= intoalist(Nphcdanigeriatweet.apply(clean))
newAPINNigeria= intoalist(APINNigeriatweet.apply(clean))
newEpiAFRIC= intoalist(EpiAFRICtweet.apply(clean))
newW4HNigeria= intoalist(W4HNigeriatweet.apply(clean))
newSFHNigeria= intoalist(SFHNigeriatweet.apply(clean))
newNCDCgov= intoalist(NCDCgovtweet.apply(clean))

data = [newnighealthwatch, newnmanigeria, newFmohnigeria, newNphcdanigeria,
newAPINNigeria, newEpiAFRIC,  newW4HNigeria, newSFHNigeria, newNCDCgov]

#Dictionary encapsulates the mapping between
#normalized words and their integer ids.
dictionary = gensim.corpora.Dictionary(data)
dictionary.save('dictionary.dict')
print( dictionary)

#Convert document (a list of words) into the bag-of-words format
#list of (token_id, token_count) 2-tuples.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data]

#The Matrix Market (MM) exchange formats provide a simple mechanism to facilitate the
exchange of matrix data
gensim.corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

model_list, coherence_values = compute_coherence_values(dictionary=dictionary,
corpus=doc_term_matrix, texts=data, start=2, limit=20, step=1)
```

```
# Show graph
limit=20; start=2; step=1;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

num_topic = 5
Lda = gensim.models.LdaMulticore#LdaMulticore uses all CPU cores to parallelize and
speed up model training
lda= Lda(doc_term_matrix, num_topics=num_topic,id2word = dictionary,
passes=20,chunksize=100,random_state=3)

lda.save("lda_topic_model")

for i in lda.print_topics():
    for j in i: print(j)

dic = gensim.corpora.Dictionary.load('dictionary.dict')
corp = gensim.corpora.MmCorpus('corpus.mm')
lda_tp = gensim.models.LdaModel.load("lda_topic_model")

viz= pyLDAvis.gensim.prepare(lda_tp, corp, dic)
viz

ldamallet = gensim.models.wrappers.LdaMallet('mallet-2.0.8/bin/mallet', corpus= corp,
num_topics= 5, id2word= dictionary)
# LDA Mallet Model
for i in ldamallet.print_topics():
    for j in i: print(j)

for t in range(ldamallet.num_topics):
    plt.figure()
    plt.imshow(WordCloud().fit_words(dict(ldamallet.show_topic(t, 200))))
    plt.axis("off")
    plt.title("Topic #" + str(t))
    plt.show()

"""# For 2017"""

# Read in CSV files into DataFrames for each twitter handle
nighealthwatch= pd.read_csv("nighealthwatch2017.csv")
nmanigeria= pd.read_csv("nmanigeria2017.csv")
Fmohnigeria= pd.read_csv("Fmohnigeria2017.csv")
Nphcdanigeria= pd.read_csv("NphcdaNG2017.csv")
APINNigeria= pd.read_csv("APINNigeria2017.csv")
EpiAFRIC= pd.read_csv("EpiAFRIC2017.csv")
W4HNigeria= pd.read_csv("W4HNigeria2017.csv")
SFHNigeria= pd.read_csv("SFHNigeria2017.csv")
NCDCgov= pd.read_csv("NCDCgov2017.csv")
wharc_nigeria= pd.read_csv("wharc_nigeria2017.csv")
WHONigeria= pd.read_csv("WHONigeria2017.csv")

#Retrieving only tweet text information for each twitter handle
nighealthwatchtweet= nighealthwatch["text"]
nmanigeriatweet= nmanigeria["text"]
```

```
Fmohnigeriatweet= Fmohnigeria["text"]
Nphcdanigeriatweet= Nphcdanigeria["text"]
APINNigeriatweet= APINNigeria["text"]
EpiAFRICtweet= EpiAFRIC["text"]
W4HNigeriatweet= W4HNigeria["text"]
SFHNigeriatweet= SFHNigeria["text"]
NCDCgovtweet= NCDCgov["text"]
wharc_nigeriatweet= wharc_nigeria["text"]
WHONigeriatweet= WHONigeria["text"]

#Pre-process tweets
newnighealthwatch= intoalist(nighealthwatchtweet.apply(clean))
newnmanigeria= intoalist(nmanigeriatweet.apply(clean))
newFmohnigeria= intoalist(Fmohnigeriatweet.apply(clean))
newNphcdanigeria= intoalist(Nphcdanigeriatweet.apply(clean))
newAPINNigeria= intoalist(APINNigeriatweet.apply(clean))
newEpiAFRIC= intoalist(EpiAFRICtweet.apply(clean))
newW4HNigeria= intoalist(W4HNigeriatweet.apply(clean))
newSFHNigeria= intoalist(SFHNigeriatweet.apply(clean))
newNCDCgov= intoalist(NCDCgovtweet.apply(clean))
newwharc_nigeria= intoalist(wharc_nigeriatweet.apply(clean))
newWHONigeria= intoalist(WHONigeriatweet.apply(clean))

data = [newnighealthwatch, newnmanigeria, newFmohnigeria, newNphcdanigeria,
newAPINNigeria, newEpiAFRIC,  newW4HNigeria, newSFHNigeria,
newNCDCgov,newwharc_nigeria,newWHONigeria]

#Dictionary encapsulates the mapping between
#normalized words and their integer ids.
dictionary = gensim.corpora.Dictionary(data)
dictionary.save('dictionary.dict')
print( dictionary)

#Convert document (a list of words) into the bag-of-words format
#list of (token_id, token_count) 2-tuples.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data]

#The Matrix Market (MM) exchange formats provide a simple mechanism to facilitate the
exchange of matrix data
gensim.corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

model_list, coherence_values = compute_coherence_values(dictionary=dictionary,
corpus=doc_term_matrix, texts=data, start=2, limit=20, step=1)
# Show graph
limit=20; start=2; step=1;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

num_topic = 3
Lda = gensim.models.LdaMulticore#LdaMulticore uses all CPU cores to parallelize and
speed up model training
lda= Lda(doc_term_matrix, num_topics=num_topic,id2word = dictionary,
passes=20,chunksize=100,random_state=3)
```

```
lda.save("lda_topic_model")

for i in lda.print_topics():
    for j in i: print(j)

dic = gensim.corpora.Dictionary.load('dictionary.dict')
corp = gensim.corpora.MmCorpus('corpus.mm')
lda_tp = gensim.models.LdaModel.load("lda_topic_model")

viz= pyLDAvis.gensim.prepare(lda_tp, corp, dic)
viz

ldamallet = gensim.models.wrappers.LdaMallet('mallet-2.0.8/bin/mallet', corpus= corp,
num_topics= 4, id2word= dictionary)
# LDA Mallet Model
for i in ldamallet.print_topics():
    for j in i: print(j)

for t in range(ldamallet.num_topics):
    plt.figure()
    plt.imshow(WordCloud().fit_words(dict(ldamallet.show_topic(t, 200))))
    plt.axis("off")
    plt.title("Topic #" + str(t))
    plt.show()

"""# For 2018"""

# Read in CSV files into DataFrames for each twitter handle
nighealthwatch= pd.read_csv("nighealthwatch2018.csv")
nmanigeria= pd.read_csv("nmanigeria2018.csv")
Fmohnigeria= pd.read_csv("Fmohnigeria2018.csv")
Nphcdanigeria= pd.read_csv("NphcdaNG2018.csv")
APINNigeria= pd.read_csv("APINNigeria2018.csv")
EpiAFRIC= pd.read_csv("EpiAFRIC2018.csv")
W4HNigeria= pd.read_csv("W4HNigeria2018.csv")
SFHNigeria= pd.read_csv("SFHNigeria2018.csv")
NCDCgov= pd.read_csv("NCDCgov2018.csv")
wharc_nigeria= pd.read_csv("wharc_nigeria2018.csv")
WHONigeria= pd.read_csv("WHONigeria2018.csv")

NCDCgov[NCDCgov.text.isnull()]
NCDCgov.drop(1111, inplace=True)

#Retrieving only tweet text information for each twitter handle
nighealthwatchtweet= nighealthwatch["text"]
nmanigeriatweet= nmanigeria["text"]
Fmohnigeriatweet= Fmohnigeria["text"]
Nphcdanigeriatweet= Nphcdanigeria["text"]
APINNigeriatweet= APINNigeria["text"]
EpiAFRICtweet= EpiAFRIC["text"]
W4HNigeriatweet= W4HNigeria["text"]
SFHNigeriatweet= SFHNigeria["text"]
NCDCgovtweet= NCDCgov["text"]
wharc_nigeriatweet= wharc_nigeria["text"]
WHONigeriatweet= WHONigeria["text"]

#Pre-process tweets
newnighealthwatch= intoalist(nighealthwatchtweet.apply(clean))
```

```
newnmanigeria= intoalist(nmanigeriatweet.apply(clean))
newFmohnigeria= intoalist(Fmohnigeriatweet.apply(clean))
newNphcdanigeria= intoalist(Nphcdanigeriatweet.apply(clean))
newAPINNigeria= intoalist(APINNigeriatweet.apply(clean))
newEpiAFRIC= intoalist(EpiAFRICtweet.apply(clean))
newW4HNigeria= intoalist(W4HNigeriatweet.apply(clean))
newSFHNigeria= intoalist(SFHNigeriatweet.apply(clean))
newNCDCgov= intoalist(NCDCgovtweet.apply(clean))
newwharc_nigeria= intoalist(wharc_nigeriatweet.apply(clean))
newWHONigeria= intoalist(WHONigeriatweet.apply(clean))

data = [newnighealthwatch, newnmanigeria, newFmohnigeria, newNphcdanigeria,
newAPINNigeria, newEpiAFRIC,  newW4HNigeria, newSFHNigeria,
newNCDCgov,newwharc_nigeria,newWHONigeria]

#Dictionary encapsulates the mapping between
#normalized words and their integer ids.
dictionary = gensim.corpora.Dictionary(data)
dictionary.save('dictionary.dict')
print( dictionary)

#Convert document (a list of words) into the bag-of-words format
#list of (token_id, token_count) 2-tuples.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data]

#The Matrix Market (MM) exchange formats provide a simple mechanism to facilitate the
exchange of matrix data
gensim.corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

model_list, coherence_values = compute_coherence_values(dictionary=dictionary,
corpus=doc_term_matrix, texts=data, start=2, limit=20, step=1)
# Show graph
limit=20; start=2; step=1;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

num_topic = 5
Lda = gensim.models.LdaMulticore#LdaMulticore uses all CPU cores to parallelize and
speed up model training
lda= Lda(doc_term_matrix, num_topics=num_topic,id2word = dictionary,
passes=20,chunksize=100,random_state=3)

lda.save("lda_topic_model")

for i in lda.print_topics():
    for j in i: print(j)

dic = gensim.corpora.Dictionary.load('dictionary.dict')
corp = gensim.corpora.MmCorpus('corpus.mm')
lda_tp = gensim.models.LdaModel.load("lda_topic_model")

viz= pyLDAvis.gensim.prepare(lda_tp, corp, dic)
viz
```

```
ldamallet = gensim.models.wrappers.LdaMallet('mallet-2.0.8/bin/mallet', corpus= corp,
num_topics= 5, id2word= dictionary)
# LDA Mallet Model
for i in ldamallet.print_topics():
    for j in i: print(j)

for t in range(ldamallet.num_topics):
    plt.figure()
    plt.imshow(WordCloud().fit_words(dict(ldamallet.show_topic(t, 200))))
    plt.axis("off")
    plt.title("Topic #" + str(t))
    plt.show()

"""# For 2019"""

#NCDCgov.drop(70, inplace=True)
# Read in CSV files into DataFrames for each twitter handle
nighealthwatch= pd.read_csv("nighealthwatch2019.csv")
nmanigeria= pd.read_csv("nmanigeria2019.csv")
Fmohnigeria= pd.read_csv("Fmohnigeria2019.csv")
Nphcdanigeria= pd.read_csv("NphcdaNG2019.csv")
APINNigeria= pd.read_csv("APINNigeria2019.csv")
EpiAFRIC= pd.read_csv("EpiAFRIC2019.csv")
W4HNigeria= pd.read_csv("W4HNigeria2019.csv")
SFHNigeria= pd.read_csv("SFHNigeria2019.csv")
NCDCgov= pd.read_csv("NCDCgov2019.csv")
wharc_nigeria= pd.read_csv("wharc_nigeria2019.csv")
WHONigeria= pd.read_csv("WHONigeria2019.csv")

#Retrieving only tweet text information for each twitter handle
nighealthwatchtweet= nighealthwatch["text"]
nmanigeriatweet= nmanigeria["text"]
Fmohnigeriatweet= Fmohnigeria["text"]
Nphcdanigeriatweet= Nphcdanigeria["text"]
APINNigeriatweet= APINNigeria["text"]
EpiAFRICtweet= EpiAFRIC["text"]
W4HNigeriatweet= W4HNigeria["text"]
SFHNigeriatweet= SFHNigeria["text"]
NCDCgovtweet= NCDCgov["text"]
wharc_nigeriatweet= wharc_nigeria["text"]
WHONigeriatweet= WHONigeria["text"]

#Pre-process tweets
newnighealthwatch= intoalist(nighealthwatchtweet.apply(clean))
newnmanigeria= intoalist(nmanigeriatweet.apply(clean))
newFmohnigeria= intoalist(Fmohnigeriatweet.apply(clean))
newNphcdanigeria= intoalist(Nphcdanigeriatweet.apply(clean))
newAPINNigeria= intoalist(APINNigeriatweet.apply(clean))
newEpiAFRIC= intoalist(EpiAFRICtweet.apply(clean))
newW4HNigeria= intoalist(W4HNigeriatweet.apply(clean))
newSFHNigeria= intoalist(SFHNigeriatweet.apply(clean))
newNCDCgov= intoalist(NCDCgovtweet.apply(clean))
newwharc_nigeria= intoalist(wharc_nigeriatweet.apply(clean))
newWHONigeria= intoalist(WHONigeriatweet.apply(clean))

data = [newnighealthwatch, newnmanigeria, newFmohnigeria, newNphcdanigeria,
newAPINNigeria, newEpiAFRIC,  newW4HNigeria, newSFHNigeria,
newNCDCgov,newwharc_nigeria,newWHONigeria]
```

```
#Dictionary encapsulates the mapping between
#normalized words and their integer ids.
dictionary = gensim.corpora.Dictionary(data)
dictionary.save('dictionary.dict')
print( dictionary)

#Convert document (a list of words) into the bag-of-words format
#list of (token_id, token_count) 2-tuples.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in data]

#The Matrix Market (MM) exchange formats provide a simple mechanism to facilitate the
exchange of matrix data
gensim.corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

model_list, coherence_values = compute_coherence_values(dictionary=dictionary,
corpus=doc_term_matrix, texts=data, start=2, limit=20, step=1)
# Show graph
limit=20; start=2; step=1;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()

num_topic = 3
Lda = gensim.models.LdaMulticore#LdaMulticore uses all CPU cores to parallelize and
speed up model training
lda= Lda(doc_term_matrix, num_topics=num_topic,id2word = dictionary,
passes=20,chunksize=100,random_state=3)

lda.save("lda_topic_model")

for i in lda.print_topics():
    for j in i: print(j)

dic = gensim.corpora.Dictionary.load('dictionary.dict')
corp = gensim.corpora.MmCorpus('corpus.mm')
lda_tp = gensim.models.LdaModel.load("lda_topic_model")

viz= pyLDAvis.gensim.prepare(lda_tp, corp, dic)
viz

ldamallet = gensim.models.wrappers.LdaMallet('mallet-2.0.8/bin/mallet', corpus= corp,
num_topics= 4, id2word= dictionary)
# LDA Mallet Model
for i in ldamallet.print_topics():
    for j in i: print(j)

for t in range(ldamallet.num_topics):
    plt.figure()
    plt.imshow(WordCloud().fit_words(dict(ldamallet.show_topic(t, 200))))
    plt.axis("off")
    plt.title("Topic #" + str(t))
    plt.show()
```