21

Retrieval of Unstructured Datasets and R Implementation of Text Analytics in the Climate Change Domain

Olusesan Michael Awoleye and Albert Ayorinde Abegunde
Obafemi Awolowo University

O. Olawale Awe
University of Campinas

CONTENTS

21.1 Introduction

The disruptive innovation we witness today in communication and information retrieval (IR) is premised on the development of emerging technologies, which, in turn, is powered by web technology and the Internet (Awoleye *et al*., 2014; Pegoraro, 2014; Vitolo *et al*., 2015). In the era of globalization, SM has transformed the way we live and has become

DOI: 10.1201/9781003261148-25

indispensable for effective communication among people, peers, families, friends, and organizations globally (Onyijen *et al*., 2019). The adoption of SM and its effective use has gone beyond personal use for communication, information exchange, and pleasures; SM is now being employed to revolutionize communication and collaboration in the scientific sphere. It was reported that Facebook has over 2 billion users globally, which is about a third of the world population (Oltulu *et al*., 2018). In the same vein, Twitter was also noted to have over 328 million users generating over 500 million tweets daily (Wasim, 2017).[1] Other platforms in this category are LinkedIn with 610 million monthly users, Pinterest with 250 million monthly visitors, and Instagram with 100 million, among others.[2]The avalanche of data generated by users on various SM platforms such as Facebook, Twitter, YouTube, Instagram, LinkedIn, Pinterest, and Flickr cannot be overemphasized. The volume of data generated on these platforms by users daily has necessitated the need to employ machine learning techniques to filter useful information from such big data. It is in this regard that this work has employed machine learning techniques to harvest and analyze climate change data in Twitter over Nigeria geolocation.

Climate change is important at this time because scenes of flood, storms, wildfire, and other natural disasters fed by global warming are indications of how much weather and climate can affect our lives. In the context of Nigeria, it is therefore important to explore how Nigeria's climate has been changing. This includes, but is not limited to, increased temperature, variable rainfall, rises in sea levels and flooding, drought, desertification, land degradation, and likely extreme weather effects that affect humans and the ecosystem. Thus, the research employed a qualitative study using opinions on Twitter as a source of data for the analysis (Kumar and Bala, 2016; Bose *et al*., 2019). In this regard, natural language processing (NLP) procedures have been used extensively to explore big data in similar research (Procter *et al*., 2013; Agerri *et al*., 2015; Baltas *et al*., 2016). It also has some related various programming implementations, including R, Python, Java, and Matlab, that are largely object-oriented. The choice of R programming for the implementation of the unstructured dataset used to illustrate text mining in this document is premised on its robustness, versatility, scalability, reproducibility, and level of community support (McMurdie and Holmes, 2013; Vitolo *et al*., 2015). It is interesting to note that these features limit the existing statistical software packages in handling the required analysis in this context. R also provides a solution to addressing these challenges using the step-by-step approach involved in text mining of small to a large dataset, as explained in this document. R as an open-source software also has support for other software libraries: packages and functions, extending the functionality of the base R language and core packages. This will further help to demystify the procedures with concise steps. In addition, it is important to reiterate the inclusion of documentation and examples, which are usually released alongside these packages, which reduces the herculean task of learning programming languages, especially for people without prior programming knowledge. The Comprehensive R Archive Network now has over 10,000 packages that are published under scrutiny for procedural conformity and interoperability. This document, thus, enunciates some related machine learning concepts as background to provide a better understanding of the R implementation.

[1] https://www.dsayce.com/social-media/tweets-day/.
[2] https://www.internetworldstats.com/social.htm.

21.2 Social Media as a Veritable Source of Data

Social media (SM) does not have a universal definition; among many definitions advanced by researchers, Kaplan and Haenlein (2010) refer to it as a virtual social world. In the same vein, Drury (2008) described SM as online resources that people use to *share* 'content': video, photos, images, text, ideas, insight, humor, opinion, gossip, news—the list goes on. These resources include blogs, vlogs, social networks, message boards, podcasts, public bookmarking, and wikis. In this document, therefore, we define the concept as emerging and disruptive technologies that provide a medium for the creation and exchange of information among a community connected by beliefs, interests, ideologies, and careers. Part of the major benefits that the emerging technologies have brought to mankind are their ability to propagate the expression of opinions, thoughts, and feelings (Ahmed, 2017) on products, processes, ideologies, or beliefs. Businesses, both public and private, and other organizations use SM to showcase or to promote their businesses and products as well as to get feedback from users and customers. As millions of users make use of these platforms daily, they generate huge amounts of data. These data are usually not in any unique form or structure that may make it easily accessible using search queries for their retrieval. These largely unstructured datasets tend to contain useful knowledge and information that may be somehow latent. This is where the relevance of machine learning has come to the fore. Apart from the fact that SM has been used widely for communication in our daily lives, it has also been employed as a tool during crises and extreme circumstances (Ahmed, 2017), such as outbreaks of infectious diseases such as bird flu, Lassa fever, Ebola, and coronavirus, and other natural disasters like hurricanes and typhoons (Takahashi *et al.*, 2015). Different SM platforms are designed to achieve different purposes of social networking. For example, Facebook is used to connect to friends and families and to discover what is going on in the world and to share and express one's own opinion on it. Unlike Facebook, which uses both text and pictures, Instagram specifically relies on visuals such as photos and videos. Twitter, in the same vein, is a microblogging technology that allows one to send and receive short messages/posts called tweets that are usually about 140 characters long and can include links, hashtags, special characters, etc. It is important to state that data from Twitter and others are used by academic researchers examining how people use the tools in different circumstances (Ahmed, 2017). For example, the recent wave currently sweeping America and other developed countries regarding the police brutality and inequality in the social and justice system in America was first promoted on SM. This started when passers-by filmed and later uploaded a video of how American police tortured and eventually killed one African-American George Floyd. This brewed rage and eventually led to several days of street protests. A hashtag #BlackLivesMatter was then created on Twitter to assist them to collate opinions and posts relating to fighting their cause for freedom. Hashtags have been used considerably in research, especially on Twitter. For example, Chae (2015) explored the hashtag #Supplychain to gain insights into the concept of supply chain management. In the same vein, the hashtags #YolandaPH and #Haiyan were used in a research carried out by Takahashi *et al.* (2015) regarding a typhoon disaster in the Philippines, while #EbolaOutbreakAlert was used to investigate the outbreak of the Ebola pandemic (Ahmed and Bath, 2015).

21.3 Web Crawling

Web crawling can be described as a program that most search engines use to find what is new on the Internet; it is sometimes referred to as a web crawler or a spider. For example, Google's web crawler is known as GoogleBot.[3] The bot crawls the web and collects documents to build a searchable index for the different search procedures. The program starts at a website and follows every hyperlink on each page. One can say that everything on the web will eventually be found and collected, because the bot hops from one website to another. In a practical sense, a list of websites to be crawled or ignored could be itemized in a text file, e.g., 'robots.txt', which is sometimes called robot exclusion protocol. This is synonymous with an access control list in computer networking, which is a file that specifies the ports, protocols, IP addresses, etc., to be allowed or disallowed within a networking environment (Ferraiolo *et al*., 1999).

21.4 Web Crawler Architectures

It is not sufficient for a web crawler to have a good strategy; it is also desirable for it to have a highly optimized architecture. The architecture shown in Figure 21.1 is a typical one of a web crawler.

Traditional web crawling employed a set of instructions provided in a given text file such as robot.txt, which are sometimes referred to as a set of policies. At the end of the

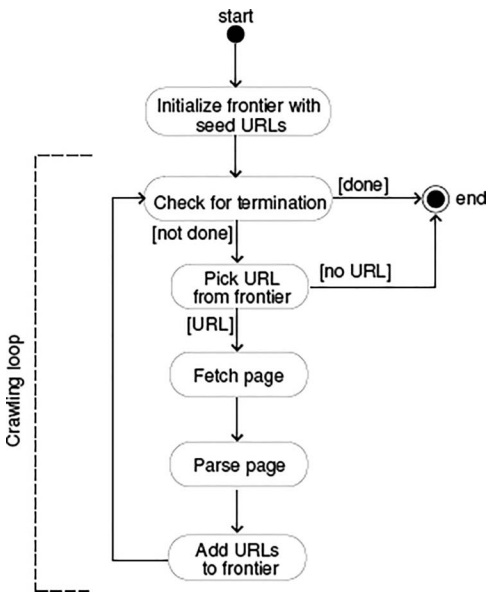

FIGURE 21.1
Crawling procedure. (Pant *et al*., 2004.)

[3] http://www.wpthemesplanet.com/2009/09/how-does-web-crawler-spider-work/ retrieved 12 June, 2010.

file, a marker is provided to show the end of the file; otherwise, the crawler moves to the next line to execute the instruction. This is recursively parsed while reaping the relevant page(s) specified, parsing them, and adding URL to the frontier. This process is recursive and continues until the end of the file is reached. It is worth stating here that Shkapenyuk and Suel (2002) reported the insistence of several scholars of the necessity of crawling important pages first. The authors further mentioned the possibility of crawling pages on a particular topic or of a particular type. Re-crawling (refreshing) pages to optimize the overall 'freshness' of a collection of pages and scheduling of crawling activities over time was also noted to be of utmost importance.

21.5 Vector-Based Model of IR

The main focus in the IR field is to be able to effectively search for information relevant to the user's needs within a given gamut of data (Oren, 2005). Several search methods are available, but one of the most popular paradigms for indexing and searching is the vector-based model of IR. One of the vector-based models is a family of variants of a very widely used scheme referred to as term frequency, inverse document frequency (*tf.idf)* methods (Salton, 1989). These schemes employ a small number of documents, collections, and query features to provide a measure of relevance for each document relative to the user's query. For example, in a recommender system, assume that N is the total number of documents that can be recommended to users and n is the total number of documents containing the index term, f is the raw term frequency, and mf is the maximum raw term frequency. Figure 21.2 shows a function that integrates these parameters as $\left(f/mf\right)X\log\frac{N}{n}$.

Since N is the total number of documents that can be recommended to users, let us also assume that the keyword k_i appears in n_i of them (Adomavicius and Tuzhilin, 2005) and $f_{i,j}$ is the number of times keyword k_i appears in document d_j. Then $TF_{i,j}$, the term frequency (or normalized frequency) of keyword k_i in document d_j, is defined as

$$TF_{i,j} = \frac{f_{i.j}}{\max_z f_{z,j}}$$

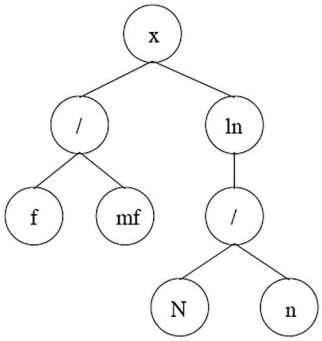

FIGURE 21.2
A parse tree of td.idf indexer. (Oren, 2002.)

where the maximum is computed over the frequencies $f_{z,j}$ of all keywords k_z that appear in the document d_j. However, keywords that appear in many documents are not useful for distinguishing a relevant document and a nonrelevant one. Therefore, the measure of inverse document frequency (IDF_i) is often used in combination with simple term frequency ($TF_{i,j}$). The inverse document frequency for keyword k_i is usually defined as

$$IDF_i - \log \frac{N}{n_i}.$$

Then the *TF-IDF* weight for keyword k_i in document d_j is defined as

$$w_{i,j} = TF_{i,j} \times IDF_i$$

and the content of document d_j is defined as Content(d_j)=(w_{1j}, …, w_{kj}).

21.6 Document Indexing

Document indexing is the process of transforming document text into a representation of text, comprising three steps: tokenization, filtration, and stemming. During tokenization, terms are lowercased and punctuation marks removed. Rules must be in place so that digits, hyphens, and other symbols can be parsed properly. Tokenization is followed by filtration. During filtration, commonly used terms and terms that do not add any semantic meaning (stopwords) are removed. In most IR systems, survival terms are further reduced to common stems or roots, this is known as stemming. Thus, the initial content of length l is reduced to a list of terms (stems and words) of length l' (i.e., $l'<l$). Raw text in a collection is first parsed into generalized words called tokens. Tokenization first lowercases the terms and includes the removal of punctuations, spaces, and other symbols that made the structure found in the article headers (e.g., CSS).

Noun phrases were used as tokens since this has been proven in the literature to be useful by Lang (1995). Thereafter, a vector of token counts for the document is created. This is the size of the total vocabulary without reckoning with tokens not occurring in the documents, which could be zero. This approach is generally known as the *bag-of-words* model and is the basis for this representation. This approach does not capture the order of the tokens in the document; it is assumed that it captures the necessary information needed for our filtering purposes. The next step is referred to as filtration, which is the process of removing the frequently occurring (used) terms sometimes called stopwords: 'the', 'of'', 'and', etc.; this is done since either their inclusion or their removal does not impact retrieval efficiency. Stopwords are filtered out before or after the processing of natural language, as they do not seem to add much meaning to a sentence when they are ignored or excluded from it.

21.6.1 Indexing Using LUCENE

Lucene uses a combination of a Vector Space Model (VSM) and a Boolean model to determine the relevance of a document to a given query. In IR, documents are represented as

vectors (Polettini, 2004) and the term weighting techniques serve as a determinant of the level of success or failure of the vector space method (model). The main idea behind the VSM is the more times a query term appears in a document relative to the number of times the term appears in all documents in the entire collection, the more relevant the document is to the query. Lucene uses the Boolean model to first narrow down the document that needs to be scored based on the use of Boolean logic in the query specification. Lucene draws its popularity and/or strength from the good results and applicability to nonstructural texts. Two issues are considered when weights are to be assigned to terms especially in the IR domain: (1) the local information from individual documents and (2) the global information from a collection of documents. Salton (1998) is a leading study on this point by presenting a VSM, commonly known as the 'term vector model'. The weighting scheme is presented as follows:

$$\left(\text{Term Weight}\right) w_i = tf_i * \log\left(\frac{D}{df_i}\right) \tag{21.1}$$

where

- tf_i = term frequency (term counts) or the number of times a term i occurs in a document.
- df_i = document frequency or the number of documents containing term i.
- D = number of documents in the database.

21.7 Local Weights

The equation shows that the weight of a term (w_i) increases with term counts (tf_i). This is observed as being vulnerable to term repetition abuses in the model; Garcia (2006) described it as an adversarial practice known as keyword spamming.

Given a query q,

1. for documents of equal lengths, those with more instances of q are favored during retrieval; and
2. for documents of different lengths, longer documents are favored during retrieval since these tend to contain more instances of q.

21.8 Global Weights

In equation (21.1), the $\log(D/df_i)$ term is known as the inverse document frequency (IDF_i), a measure of the volume of information associated with term i within a set of documents. Inspecting the df_i/D ratio, this is the probability of retrieving from D a document containing term i. The probability is simply inverted in the equation and the log is taken. The result is then pre-multiplied by tf_i.

21.9 Methods of Accessing Datasets Online

This section describes four main techniques for retrieving data into the R environment: (1) using open-source and proprietary software, (2) directly using the application program interface (API) with some R codes, (3) pointing directly to the data via a given web page, and (4) using files from other sources such as comma-separated files (CSV) or text files. Before this, the first thing that must be done is setting the RStudio environment; this includes getting and setting the working directory. This makes working with files much easier when one intends to retrieve or save a given file from/to the local drive. The following R codes could be used to achieve this.

```
# Set current working directory.
setwd("/Analysis/climate") #the parameters in parentheses represents the
file path

# Get and print current working directory.
getwd()
```

21.9.1 Using Open-Source and Proprietary Software

There are many ways in which data could be crawled online and fetched into the R programming environment. One such way is by using Twitter Archiving Google Sheet as designed by Hawksey (2014). This is a Google spreadsheet that only requires the users to specify the keyword, hashtags, or combination of hashtags to be retrieved. Logical operations such as 'OR' or 'AND' are allowed; this searches for either of the keywords/hashtags or combines the given search terms in the query, respectively. There are other open-source software that can be used to fetch data from Tweeter as well, such as Mozdeh and Chorus. Other proprietary software includes NodeXL, RapidMiner, and Discover Text. These require Twitter API (Kim *et al*., 2020), which necessitates that a user registers a Twitter account and requests the authorization code, which consists of consumer_key and the consumer_secret key. These are computer-generated series of alphanumeric codes.

21.9.2 Retrieving Data from Twitter Using API

```
#This section installs and load the twitter package to
#the RStudio environment
          install.packages("twitteR")
          library(twitteR)

#This sets up the API permission procedures which must
#be generated from a twitter account
consumer_key<- "PxwadCtG0tGdDlAe2Y18yXqaZ"
consumer_secret<- "fQpST1zaYXhMby3CH19LKZdYVO60R47v1P4wevKLnoBGSGQAWe"
access_token<- "45821565-MH8DOws1M3ol6chaKEX9nGOn5Oa7ceSvhYwtWcVzM"
access_secret<- "3RbCpTkMrGdCX7ewMmMhhaII62WAkVrL6qnZZueVqoBJe" setup_
twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

# The following line of codes requests 2000 tweets which
# contains #ClimateChange hashtag
```

	text	replyToSN	created	truncated	retweetCount	isRetweet	retweeted	longitude	latitude
1	RT @CanadaFP: Minister Champagne spoke to Foreign Minister Casten Nemra of the #M	NA	11/06/2020 0:42	FALSE	1	TRUE	FALSE	NA	NA
2	RT @AnikaMolesworth: We must transition quickly & effectively away from	NA	11/06/2020 0:42	FALSE	19	TRUE	FALSE	NA	NA
3	RT @almacardi: Top story: @Hana_ElSayyed: 'power paint	NA	11/06/2020 0:41	FALSE	11	TRUE	FALSE	NA	NA
4	RT @DrJackiSmall: Our lives depends on it #ClimateChange https://t.co/mX6C2jzTCv	NA	11/06/2020 0:41	FALSE	2	TRUE	FALSE	NA	NA
5	Minister Champagne spoke to Foreign Minister Casten Nemra of the #MarshallIslands to	NA	11/06/2020 0:41	TRUE	1	FALSE	FALSE	NA	NA
6	Outside between a full afternoon and evening of events. The wind is warm and strong.	NA	11/06/2020 0:41	TRUE	0	FALSE	FALSE	NA	NA
7	RT @GerberKawasaki: Dear @elonmusk and @Tesla - your sustainability report was ama	NA	11/06/2020 0:41	FALSE	313	TRUE	FALSE	NA	NA
8	The world knew, in a sense, this was coming. COVID-19 is Disease X, according to ANZ ch	NA	11/06/2020 0:40	TRUE	0	FALSE	FALSE	NA	NA
9	#2018 #Conference - #author #AndrewKimbrell; has been a leading proponent of regene	NA	11/06/2020 0:40	TRUE	0	FALSE	FALSE	NA	NA
10	RT @LowyInstitute: Leadership from big-city mayors, CEOs, and others on climate chang	NA	11/06/2020 0:38	FALSE	1	TRUE	FALSE	NA	NA
11	RT @saveearth1928: A TOTALLY #TRANSPARENT TOTALLY #EDUCATION & #EQUALITY	NA	11/06/2020 0:38	FALSE	1	TRUE	FALSE	NA	NA
12	RT @AnikaMolesworth: We must transition quickly & effectively away from	NA	11/06/2020 0:37	FALSE	19	TRUE	FALSE	NA	NA
13	RT @PercievedLogic: A TOTALLY #TRANSPARENT TOTALLY #EDUCATION & #EQUALIT	NA	11/06/2020 0:37	FALSE	1	TRUE	FALSE	NA	NA
14	RT @arikring: @cberrl @DawnRoseTurner @C37H42Cl2N2O6 @RuthPtn @ZacharyPBeasl	NA	11/06/2020 0:37	FALSE	357	TRUE	FALSE	NA	NA
15	RT @AnikaMolesworth: We must transition quickly & effectively away from	NA	11/06/2020 0:36	FALSE	19	TRUE	FALSE	NA	NA
16	RT @blairpalese: The #climatechange induced #bushfire season is on in the northern he	NA	11/06/2020 0:36	FALSE	1	TRUE	FALSE	NA	NA
17	RT @UNinIndia: Though it can be challenging to think beyond immediate recovery amids	NA	11/06/2020 0:36	FALSE	2	TRUE	FALSE	NA	NA
18	RT @alvinfoo: Iceberg falls off the cliff, extremely rare capture clip. #ClimateChange htt	NA	11/06/2020 0:36	FALSE	62	TRUE	FALSE	NA	NA
19	RT @AnikaMolesworth: We must transition quickly & effectively away from	NA	11/06/2020 0:36	FALSE	19	TRUE	FALSE	NA	NA
20	RT @UNinIndia: Over 100,000 people around the world think that #ClimateChange and e	NA	11/06/2020 0:36	FALSE	4	TRUE	FALSE	NA	NA

FIGURE 21.3
Raw sample Tweets.

```
#twiits = searchTwitter("#ClimateChange ", n = 2000, lang="en",
locale="Nigeria") ClimateChange<- searchTwitter("#ClimateChange", n=2000,
lang = "en", locale = "Nigeria") #the funcion twListToDF takes a list of
objects from a single
#twitteR class and return a data frame version of the members
ClimateDAta<- twListToDF(ClimateChange)
ClimateDAta
write.csv(ClimateDAta, "climateChange.csv")#this writes the dataset to disk
```

The tweets containing the #ClimateChange data frame, thus, show that 2,000 elements were harvested. This is then converted to a data frame ClimateDAta of 2,000 cases with 16 variables. Part of the output is as shown in Figure 21.3.

21.10 Retrieving Data from a Given Web Page

The following web page consists of a dataset of a 750×5 array of prison record; this consists of 750 cases with 5 attributes. The following R code, thus, scrapes this dataset from the web page into the object, christened URL as follows.

```
URL <-'https://raw.githubusercontent.com/guru99-edu/R-Programming/
master/prison.csv' #this is a single line from the previous line
myData <- read.csv(URL)[1:5]
myData            #to view/print the content retrieved from the
                     #internet/URL
str(myData)          #to check the structure of the dataset
head(myData, 5)  #to view the first five rows on top
tail (myData, 5) # to view the trailing/last five rows

#TO SAVE THE RETRIEVED DATASET TO A FILE ON YOUR COMPUTER
#SYNTAX, write.csv (source file, 'destination file')
write.csv (myData, "prison.csv")
```

21.11 Retrieving a Local CSV file

Since textual data can be stored in any file format such as CSV, TXT, JSON, RDA, PDF, HTML, or XML, data could be retrieved from a file of any type. Although R natively has support to read CSV and TXT, additional format-specific packages are required to process other file formats (Welbers, 2017). As this approach of working with various packages to access different file formats may be a herculean task, a convenient task to handle this is through a package called 'readtext' as per Welbers (2017). In this context, the 'read.csv' function will be employed to achieve the purpose of this section.

```
install.packages("readr")
 library(readr)
ClimateDAta <- read.csv ("climate.csv", header =TRUE, sep = ", ",
stringsAsFactors = FALSE)
```

Another option is to choose a given file from a local drive, this must of course support the file format of the function used, in this case, read.csv. The following R code could be used to load the file. As the code executes, it will display a dialog box allowing the user to navigate to any given file on the local drive(s).

```
ClimateDAta<- read.csv(file.choose(), sep = ", ", stringsAsFactors =
FALSE, header = TRUE)
```

21.12 Data Preparation

21.12.1 Data Cleaning

The next procedure is to select the required field (variable) from the dataset, which in this case is the tweets, represented by 'text'. The following procedure shows the R implementation to reap the tweets from the dataset.

```
climateText<- paste(ClimateDAta$text, collapse = " ")
climateText
```

Looking at the data harvested on climate change, it is very obvious that there is the need to clean up unnecessary characters and links associated with SM, which may not be useful for text mining analytics. This process is called data cleaning; there is a way to carry this out in R, which the following codes show how to do.

```
# The first stage is to install and load the required packages
install.packages("stringi") #install package
install.packages("tm")
library(stringi) #load package
library(tm)

#This section cleans the data of unnecessary characters
climateClean<- stri_replace_all(climateText, "", regex = "<.*?>")
climateClean<- stri_trim(climateClean)
#creating Vector source and Corpus
```

```
climateVSource<- VectorSource(climateClean)
ClimateCorpus<- VCorpus(climateVSource)

#########################################################################
#Removal of urls, special characters and numbers
ClimateCorpus<- tm_map(ClimateCorpus, removePunctuation)
ClimateCorpus<- tm_map(ClimateCorpus, removeNumbers)
hashtagRemoval <- function(x) gsub("#\\S+", "", x)
HandleRemoval <- function(x) gsub("@\\S+", "", x)
shortWordRemoval <- function(x) gsub('\\b\\w{1,5}\\b',' ', x)
urlRemoval <- function(x) gsub("http:[[:alnum:]]*", "", x)
ClimateCorpus<- tm_map(ClimateCorpus, content_
transformer(hashtagRemoval)) ClimateCorpus<- tm_map(ClimateCorpus,
content_transformer(HandleRemoval)) ClimateCorpus<- tm_map(ClimateCorpus,
content_transformer(shortWordRemoval)) ClimateCorpus<- tm_
map(ClimateCorpus, content_transformer(urlRemoval))
ClimateCorpus<- tm_map(ClimateCorpus, removeWords, c("ClimateChange",
"httpstcoEKIZYxdhg", "\U0001f30f", "…"))
#########################################################################

# TOKENIZATION AND STEMMING
install.packages("quanteda")
 library(quanteda)
ClimateCorpusChar<- corpus(ClimateCorpus) #reverting the treated corpus #
               #back to character
climateToken<- tokens(ClimateCorpusChar) #tokenise to unigram
climateToken<- tokens_tolower(climateToken)
climateToken <- tokens_wordstem(climateToken)
 stopWds<- stopwords("english")
climateTokenNoSTOP<- tokens_remove(climateToken, stopWds)#stopwords
removal
# Weighting using dfm
dtm<-dfm(climateTokenNoSTOP) dtm<- dfm_remove(dtm, c('#*', '@*'))
dtm<- dfm_remove(dtm, c('rt', 'amp'))
#THE FOLLOWING REMOVES UNWANTED WORDS IN THE CORPUS
dtm<- dfm_remove(dtm, c('climatechang',' climat', 'chang',' …'
,' "',' httpstcowpajmb',' gerberkawasaki',' geraldkutney',' 
pauledawson',' ayanaeliza',' elonmusk',' profstrachan',' gretathunberg',' 
thunberg',' energicaus',' -
'))
frequency <- colSums(dtm)
frequency <- sort(frequency, decreasing=TRUE) head(frequency)
dtm2<-dtm[, frequency>=4] dtm3<-dfm(dtm2)

#WORDCLOUD
textplot_wordcloud(dtm, min_size = 1.2, max_size = 4, min_count = 35,
max_words = 150, color
= "darkblue", font = NULL, rotation = 0.1)
```

21.13 Frequent Terms and *#climatechange* Discourse

One of the main features of text mining is the frequency terms, showing how frequently each term occurs in a given corpus. When terms appear frequently in a given corpus, they

FIGURE 21.4
Wordcloud showing frequent terms.

are likely to provide insights on the subjects of interest. This has been used widely in literature and remains a topical approach today since text mining in itself is a new area of NLP (Dayeen *et al.*, 2020).

In the context of tweets with the climate change hashtag, Figure 21.4 presents the wordcloud. This shows the most frequent related words used with the subject, which include 'sustain', 'impact', 'report', and 'environ'. The word 'sustain' was stemmed from words like 'sustainable', 'sustainability', 'sustained', and 'sustains'. Within the raw data, this reveals the agitation and advocacy of sustainable measures to mitigate the effect of climate change, which is important in Nigeria, where limited resources could be a barrier to appropriate measures. In the same vein, the impact of climate change on the 'environment' is also noted to be a matter of great concern. The wordcloud in Figure 21.4 further provides information about the factors contributing to environmental degradation as codified by 'environment'. Some of these are 'pollut', 'globalwarm', 'emiss', 'deforest', and 'burn'. Further inspection of the raw data shows that pollution from emissions, bush burning, fossil fuel, and deforestation characterizes the issues around climate change in Nigeria's space. Topic models have been designed and used in NLP, which has the propensity to help categorize related words under relevant features. Some of these models are the joint sentiment topic model (Lin and He, 2009) and latent Dirichlet allocation, instituted by Blei, Ng, and Jordan (2003).

21.14 Sentiment Analysis

This section describes the pattern of the tweets on climate change; this shows the polarity of the tweets as *positive*, *negative*, or *neutral* opinions. The procedure powering this underlying feature uses algorithms that combine text analyses, computational linguistics, and NLP to understand the sentiments of the words in the tweets collected. The following procedure is an R programming implementation to plot the sentiment visualizations presented in this work using the syuzhet package in R.

```
# GRAPH PLOTS
library(syuzhet)
s_v <- get_sentences(climateText)
s_v_sentiment <- get_sentiment (s_v, method="bing")
plot(s_v,
                    type="h", #other option here is "l"
                    "h" main="Climate Change Tweets",
                    xlab="Narrow Time",
                    ylab="Emotional Valence",
                    col="blue")
```

21.15 The Trajectory of Climate Change Discourse

The polarity of the opinions of the respondents as revealed in Figure 21.5 suggests mixed opinions, which range between two equal ends of emotional valence. Some were optimistic that the effects of climate change vis-à-vis the present situation can be put under control, while others expressed fear that the effects may be disastrous. In a situation like this, it is important to compare what other countries of the world, including developing economies, emerging nations, and most especially advanced countries, had done. Adopting best practices for mitigating the effect of climate change seems to be the way forward for developing countries like Nigeria.

21.16 Conclusion

The Third Industrial Revolution was characterized by the advent of many Internet technologies. The Internet technologies themselves, thus, provided the platform by which the emerging technologies thrive. Today, our way of life has been perturbed by the use of SM and its related technologies. This has been beneficial to humankind, especially in the way we communicate, interact, and share information. Despite the ubiquitous benefits that this

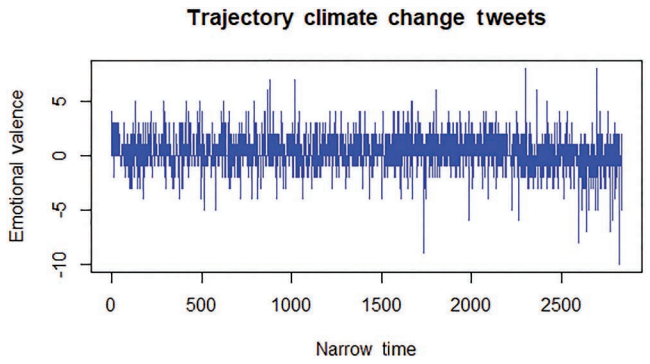

FIGURE 21.5
Trajectory of climate change tweets.

has brought to us, the challenge remains of the avalanche of data being generated daily by the use of these technologies. Improper management of this has been noted to have created bottlenecks in the way information is gathered and presented for decision-making in organizations. This is what necessitates a body of knowledge on IR in efforts to continually seek strategies and methods to retrieve useful information in the most effective way. This chapter, therefore, contributes to these efforts by using the NLP techniques via R implementation to mine the characteristics of useful information within the domain of climate change as it relates to the Nigerian cyberspace.

References

Adomavicius, G., Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734–749.

Agerri, R. Artola, X., Beloki, Z., Rigau, G., Soroa, A. (2015). Big data for natural language processing: a streaming approach. *Knowledge-Based Systems*, 79, 36–42. https://doi.org/10.1016/j.knosys.2014.11.007.

Ahmed, W. (2017). Amplified messages: how hashtag activism and Twitter diplomacy converged at #Thi-sIsACoup – and won. In Veneti, A., Reilly, P., and Atanasova, D., eds. *Politics, Protest, Emotion: Interdisciplinary Perspectives. A Book of Blogs*, 109–114. Sheffield: University of Sheffield Information School.

Ahmed, W., Bath, P. (2015). The Ebola epidemic on twitter: challenges for health informatics. *Proceeding of 17th International Symposium on Health Information Management Research*, York, UK. Available on http://eprints.whiterose.ac.uk/87728/2/Abstract%2034.pdf.

Awoleye, O.M., Ojuloge, B., Ilori, M.O. (2014). Web application vulnerability assessment and policy direction towards a secure smart government. *Government Information Quarterly*, 31(1), S118–S125,

Baltas, A., Kanavos, A., Tsakalidis, A. (2016). An Apache Spark implementation for sentiment analysis on Twitter data. In *Proceedings of the International Workshop on Algorithmic Aspects of Cloud Computing (ALGOCLOUD)*, Aarhus, Denmark.

Blei, D., Ng, A.Y., Jordan, M.I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022. Available on http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf.

Bose, R., Dey, R.K., Roy, S., Sarddar, D. (2019). Analyzing political sentiment using Twitter data. In *Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies*, 107.

Chae, B.K. (2015). Insights from hashtag #supplychain and Twitter analytics: Considering Twitter and Twitter data for supply chain practice and research. *International Journal of Production Economics*, 165, 247–259.

Dayeen, F.R., Sharma, A.S., Derrible, S. (2020). A text mining analysis of the climate change literature in industrial ecology. *Journal of Industrial Ecology*, 24(2). Available on https://doi.org/10.1111/jiec.12998.

Ferraiolo, D., Barkley, J., Kuhn, R. (1999). A role-based access control model and reference implementation within a corporate Internet. *ACM Transactions on Information and System Security*, 2(1), 34–64.

Glen, D. (2008). Opinion piece: social media: should marketers engage and how can it be done effectively? *Journal of Direct, Data and Digital Marketing Practice*, 9, 274–277.

Kaplan, A., Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of social media. *Business Horizons*, 53(1), 59–68.

Kim, Y., Nordgren, R., Emery, S. (2020). The story of Goldilocks and three Twitter's APIs: a pilot study on Twitter data sources and disclosure. *International Journal of Environmental Research and Public Health*, 17, 864. https://doi.org/10.3390/ijerph17030864.

Kumar, M., Bala, A. (2016). Analyzing Twitter sentiments through Big Data. *3rd International Conference on Computing for Sustainable Global Development (INDIACom'16)*. IEEE, New Delhi.

Lang, K. (1995). NewsWeeder: learning to filter Netnews. *Proceedings of ICML-95, 12th International Conference on Machine Learning*, Tahoe, CA, pp. 331–339.

Lin, C., He, Y. (2009). Joint sentiment/topic model for sentiment analysis CIKM '09: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, November 2009, pp. 375–384. https://doi.org/10.1145/1645953.1646003.

McMurdie, P.J., Holmes, S. (2013). phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One*, 8(4), e61217. https://doi.org/10.1371/journal.pone.0061217.

Oltulu, P., Mannan, R., Gardner, J.M. (2018). Effective use of Twitter and Facebook in pathology practice. *Human Pathology*, 73, 128–143.

Onyijen, O.H., Awoleye, O.M., Olaposi, T.O. (2019). Effectiveness of social media platforms for product marketing in Southwestern Nigeria: a firm-level analysis. *International Journal of Development and Management Review*, 14(1), 175–192.

Oren, N. (2002). Reexamining tf.idf based information retrieval with genetic programming. *Proceedings of the 2002 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology (SAICSIT)*, pp. 224–234.

Pant, G., Srinivasan, P., Menczer, F. (2004). *'Crawling the Web', Web Dynamics: Adapting to Change in Content, Size, Topology and Use*, pp. 153–178. Berlin: Springer-Verlag.

Polettini, N. (2004). The vector space model in information retrieval-term weighting problem. *Entropy*, 1–9. Available on http://citeseerx.ist.psu.edu/viewdoc/download?doi= 10.1.1.104.3479&rep=rep1&type=pdf.

Procter, R., Vis, F., Voss, A. (2013). Reading the riots on Twitter: methodological innovation for the analysis of big data. *International Journal of Social Research Methodology*, 16(3), 197–214.

Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA: Addison Wesley.

Shkapenyuk, V., Suel, T. 2002. Design and implementation of a high-performance distributed web crawler. *Proceedings of the International Conference on Data Engineering*. Available on citeseer.ist.psu.edu/shkapenyuk02design.html.458,530,531.

Takahashi, B., Tandoc, E.C., Carmichael, C. (2015). Communicating on Twitter during a disaster: an analysis of Tweets during Typhoon Haiyan in the Philippines. *Computers in Human Behavior*, 50, 392–398.

Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C.J.A., Buytaert, W. (2015). Web technologies for environmental big data. *Environmental Modelling & Software*, 63, 185–198, https://doi.org/10.1016/j.envsoft.2014.10.007.