TWITTER TRENDS ANALYSIS USING STRUCTURAL TOPIC MODELLING

BY

ALEX MUIRURI MWANGI

A RESEARCH PROJECT SUBMITTED TO THE FACULTY OF COMPUTING AND INFORMATION MANAGEMENT IN PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN DATA ANALYTICS, KCA UNIVERSITY

NOVEMBER, 2019
DECLARATION

I declare that this research project is my original work and has not been previously published or presented elsewhere for the award of a degree. I also declare that this work contains no material written or published by other people except where due reference is made and the author duly acknowledged.

Sign: ……………………………………….. Date: ……………………………

ALEX MWANGI,

Reg, No: 16/03044.

This research project has been submitted for examination with my approval as university supervisor.

Sign: ……………………………………….. Date: ……………………………

Dr. LUCY W. MBURU,

Supervisor.

KCA UNIVERSITY.
ABSTRACT

Social Networking Sites (SNS) such as Facebook and Twitter have become indispensable for netizens all over the world. They are an important source of information and entertainment for many users. Everyday increasing amounts of data is generated on these sites. This data is mostly comprised of unstructured text data (Talib, Hanif, Ayesha, & Fatima, 2016). Extracting useful information from this data would be tedious and time consuming. Humans are also error prone and can be affected by biases while computers are only influenced by the data. Computer assisted text analysis can help humans analyze this data much faster by automating the process (Talib et al., 2016). This includes techniques such as calculating the word frequency, sentiment analysis, text classification and topic modelling.

This study will implement topic modelling to extract useful topics from Twitter data. Topic modelling helps us understand what a certain text corpus is talking about. It does this by structuring and organizing the data according to word co-occurrence in different documents and grouping the words into different topics. The output of the model helps us understand the most probable topics for a particular text and can be used to classify similar but previously unseen text. This study will explore how to obtain data from Twitter application programming interface (API) and the various natural language processing (NLP) techniques that will be used to prepare the text for analysis. This study will also explore the various text modelling algorithms and determine the most appropriate one for our data. Finally topics will be estimated from the text and use various visualizations to understand and evaluate the topics.

Keywords: Twitter, topic modelling, text, analysis, API, NLP
ACKNOWLEDGEMENT

Special thanks to the almighty God for the gift of life and health.

Special thanks to my parents Mr. and Mrs. Paul Mwangi for their moral and financial support throughout this course.

I would also like to thank my supervisor Dr. Lucy W. Mburu for the guidance and assistance for the duration this project.

MAY THE LORD BLESS YOU.
DEDICATION

Dedicated to my family for their support for the duration of this course.
TABLE OF CONTENTS

DECLARATION.............................................................................................................................. ii
ABSTRACT ................................................................................................................................. iii
ACKNOWLEDGEMENT............................................................................................................... iv
DEDICATION............................................................................................................................ v
TABLE OF CONTENTS............................................................................................................... vi
LIST OF ACRONYMS AND ABBREVIATIONS ................................................................. ix
Application Programming Interface ..................................................................................... ix
Singular Value Decomposition ............................................................................................... ix
CHAPTER ONE ......................................................................................................................... 1
INTRODUCTION ...................................................................................................................... 1
  1.1 Background of the Study ................................................................................................. 1
Text analysis methods .......................................................................................................... 3
  1.2 Statement of the Problem ............................................................................................. 4
  1.3 Motivation of the Study ............................................................................................... 5
  1.4 Main Objective .............................................................................................................. 6
  1.5 Specific Objectives ........................................................................................................ 6
  1.6 Research Questions ...................................................................................................... 6
  1.7 Significance of the Study ............................................................................................. 7
  1.8 Scope of the Study ....................................................................................................... 8
CHAPTER TWO ......................................................................................................................... 9
LITERATURE REVIEW ........................................................................................................... 9
  2.1 Introduction ................................................................................................................... 9
    2.2.1 Evolution of Topic Models ...................................................................................... 9
    2.2.2 LDA ....................................................................................................................... 9
    2.2.3 Computational Methods ....................................................................................... 11
# LIST OF ACRONYMS AND ABBREVIATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag of Words</td>
</tr>
<tr>
<td>CTM</td>
<td>Correlated Topic Model</td>
</tr>
<tr>
<td>DTM</td>
<td>Document Term Matrix</td>
</tr>
<tr>
<td>LDA</td>
<td>Latent Dirichlet Allocation</td>
</tr>
<tr>
<td>LSA</td>
<td>Latent Semantic Analysis</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>POS</td>
<td>Part-of-Speech</td>
</tr>
<tr>
<td>NER</td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td>AFP</td>
<td>Agence France Presse</td>
</tr>
<tr>
<td>TM</td>
<td>Topic Modelling</td>
</tr>
<tr>
<td>TI-IDFS</td>
<td>Term Frequency – Inverse Document Frequency</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>STM</td>
<td>Structural Topic Model</td>
</tr>
</tbody>
</table>
CHAPTER ONE
INTRODUCTION

1.1 Background of the Study

Since the invention of the internet over two decades ago, the world has experienced an unprecedented increase in the amount of information generated. Most of this information is in text format. Exploring this data manually would require much effort and would be too time consuming. This has led to a great deal of research in the domain of text mining to assist users in gaining insights from the ever-growing textual data. Topic Modelling (TM) is a text mining method that has gained prominence in recent years. Topic modelling has shown to be able to give insights on huge corpus of textual data and hence improving exploration of unknown data, as an alternative to a traditional search engine.

Topic modelling was developed as an alternative to keyword search to enhance the exploration of text data collections (Deerwester, Furnas, Landauer, & Harshman, 1990). TM derives latent topics and patterns from textual data. TM has proven effective in summarizing large amounts of information and has been proposed as a solution to make long conversations, like microblogs, more approachable. This statistical model consists of the topics that appear in the data presented as a group of keywords sorted in their influence in forming the topic. TM also contain probabilities of each topic occurring in each of the document that can be used to filter all posts containing a particular topic.
Social Networking Sites (SNS) like Facebook and Twitter are a recent phenomenon that has transformed many aspects of our daily lives. Every second an average of 6000 Tweets are posted; this translates to about 500M Tweets per day.

Twitter has become an indispensable source of news and information for a wide variety of users. People use Twitter to gather real-time news, follow events of interest and read updates by people they follow. The social networking site has become important for broadcasting breaking news and eyewitness accounts. Users also utilize the site for product marketing, product reviews and dissemination of news.

Twitter users have are used to receiving real-time updates on important local and global events. For example, Twitter was utilized to distribute information in many emergency and crisis situations such as the terrorists attack on the Dusit Complex in Kenya and the aftermath of the Kenyan elections in 2017. In addition, many companies and famous personalities use their Twitter accounts to keep in touch with clients and fans.

Being a mainstream Social Networking Site, Twitter offers researchers numerous prospects for research in text mining and Natural Language Processing (NLP)(Benhardus & Kalita, 2013). One such aspect is trending topics, highlighted on Twitter’s home page. They represent what is currently popular in users’ tweets. Studying the characteristics and content of these tweets is important to aid in important research such as detection of breaking news, recommending personalized messages, recommending friends, sentiment analysis among others.

It is important to analyze the huge amount of social media data generated daily to obtain meaningful information especially during any crisis and emergency situations.
Topic Models are adept in summarizing, exploring and indexing large text document collections and can be used for this purpose (Manna, S., & Phongpanangam, O. (2018)).

**Text analysis methods**

Word Frequency: Word frequency can be used to find the most commonly occurring word in a text corpus. This is a basic text analysis technique to that can be used to show the most frequent words in a document.

Collocation: Collocation is used to find words that co-occur regularly. This is done by identifying bigrams which are two contiguous words or trigrams which are three contiguous word. These are the most frequent collocations.

Sentiment Analysis: Sentiment analysis is used to detect emotions in a certain text. It can be used to categorize emotion as either negative, neutral or positive. This helps in identifying the general mood of a certain text. This can be useful for companies for example in assessing sentiments about the company in social media.

Language Detection: Language detection is used to identify the language a certain text has been written in. It can be used to in social media, for example Twitter, to detect language and recommend posts according to the language of the user.

Keyword Extraction: Keyword extraction is used to find the most pertinent words in a certain text that can be used to summarize the text. It can be used to index or tag textual data.
Named Entity recognition: Named Entity recognition finds entities in a certain text, which be either places, organizations or people. This can be useful in finding events, locations or persons of interest in the text.

Summary Extraction: Summary extraction is used abstract large texts with minimal loss of meaning.

Word Sense Disambiguation: Word Sense Disambiguation is used find the sense of words for words with more than one meaning. For example the word ‘fly’ could mean an insect or the action of flying depending on the context it is used in a document.

Clustering: Text clustering is an unsupervised machine learning technique used to group large amounts of text according to a set of similar words that appear in the text.

Topic Modelling: Topic modelling is used to understand the discourse of a certain text. It does this by finding latent topics by modelling the distribution of topics over documents and words over the topics.

This study will focus on implementing a topic model to Twitter trending topics due to the factors discussed below in the statement of the problem.

1.2 Statement of the Problem

Microblogging sites such as Twitter restrict the number of characters that a post can contain. For example, Twitter restricts the length of a post to between 140 and 280 characters. Due to this, microblogged messages have unconventional syntax and structure. The magnitude of Twitter means that it can create a dynamic corpus. Due to these factors Twitter data presents several challenges not present in traditional analysis.
Twitter messages are often characterized by the use of obscure Language and grammar because users often omit proper punctuation and use improper grammar in their posts. Tweets usually include shortened words and URLs, abbreviations and informal lingo such as “IRL” for “in real life”. The messages are also short and hence they contain very little grammatical structure. In addition, the messages usually allude to diverse and specific events and locations and thus pre-defined entity recognition methods cannot be used.

In this study, I will investigate the use of Topic models for analyzing Twitter data. I propose that topic models are especially appropriate for analyzing Twitter data for various reasons. Firstly, topic models rely on the “bag-of-words” assumption. This means that they discard the word order and syntactic structure in the language. This makes them particularly suited to handle the improper grammar and obscure language contained in Twitter posts. Topic models can also infer latent (hidden) meanings in the data. This makes them sturdier in handling acronyms, slang and other idiosyncrasies in Twitter posts. The output of topic models are numerical vectors such as probability distributions. This makes them suitable for analysis, visualization as well advanced machine learning like clustering. Lastly, topic models are unsupervised algorithms. This makes them easily retrain able on other text data for a particular domain.

In this study, we tackle the problem of deriving a topic model from Twitter trending topics. The study gap that this study aims to fill is the inclusion of arbitrary metadata in deriving topic models that will be implemented through the structured topic model.

1.3 Motivation of the Study
Twitter has become an essential communication tool for diverse people across the world. Important events in the society are increasingly found in the timelines of individual people in Social Networking Sites such as Twitter. Trending topics utilize social media to provide a snapshot of topics and issues currently popular with users in the online community.

Researchers wish to use social media to infer users’ interests, model complex social relationships, follow news stories and identify developing topics. Companies want to use the messages posted to gain a competitive advantage, learn from their customers, better target marketing efforts and infer customers’ sentiment. Topic models are powerful algorithms to understand hidden patterns in the messages.

1.4 Main Objective

The research proposes to implement a topic model of Twitter trending topics.

1.5 Specific Objectives

(i) To investigate how topic models can be applied to analyze Twitter trending topics and establish the appropriate text preprocessing techniques to apply to Twitter messages.

(ii) To apply a topic model to Twitter trending topics.

(iii) To test and validate the effectiveness of the topic model in identifying the relevant topics from Twitter trending topics.

1.6 Research Questions
(i) How can topic models be applied to analyze Twitter trending topics and what are the appropriate text preprocessing techniques that should be applied to Twitter messages?

(ii) Which is the appropriate topic model to apply to Twitter trending topics.

(iii) How effective is the topic model in identifying the relevant topics.

1.7 **Significance of the Study**

The findings made in this study will be significant to several stakeholders:-

Traditionally, most social scientists have used either human coding or dictionary methods that require high levels of pre-analysis making them very expensive. This problem is aggravated by the ever-increasing volume and variety of unstructured text. Adopting computer-assisted methods like topic modelling would augment and amplify their social science analysis.

Companies can benefit greatly from this study because their customers usually use Twitter to voice their sentiments as in the case of product launches and complaints or compliments about a product. Topic modelling can help them sieve through a huge amount of posts in a short time to discover which topics their customers are discussing.

People use Twitter to get updates and post information during crisis situations and events of national significance. The user posts contain valuable information about events, places and people of interest. This research can help the government extract this valuable information in a short time that is critical especially in emergencies.

The study will benefit other scholars interested in topic modelling of short texts. They will know the best preprocessing techniques and the optimal algorithms to apply in these situations.
1.8 Scope of the Study

The motivation of this study is to derive topics from Twitter posts on a trending topic. The data will be obtained from tweets sampled from the Twitter search API.
CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

This chapter evaluates previous literature produced in the area of study. The review will be conducted in reference to the specific objectives of the study. The major topics that will be covered in this review are the theoretical review, empirical review and the conceptual framework.

2.2 Theoretical review

2.2.1 Evolution of Topic Models

Topic models find hidden topics within a collection of text documents. They are probabilistic models for discovering the hidden structure of a corpus of documents based on a Bayesian analysis of the documents (M. Rosen-Zvi, Griffiths, Steyvers, & Smyth, 2004). Latent Dirichlet Allocation (LDA) (D. M. Blei et al., 2003) is one of the widely used topic model today.

2.2.2 LDA

There are two main methods of computerized text analysis: statistical-based methods such as topic models (Hofmann, 2001) and natural language processing (NLP). NLP models perform Part-Of-Speech (POS) tagging, Named Entity Recognition (NER) and semantic labelling of documents. In contrast, statistical-based method use the “bag of words” (BoW) assumption. In doing this, word order and semantic structure in documents are ignored.

By ignoring the word order in documents, BoW models do not perform well short texts like question-and-answer. However, for large text corpus, the BoW assumption
provides a wider range of statistical algorithms by the assumption exchangeability, i.e., the word order does not influence the outcome of the model. (D. M. Blei & Lafferty, 2009). This assumption aids statistical-based methods in identifying semantic themes in collections of related documents.

One of the earliest applications of topic models was reducing dimensionality of large text corpora. (Deerwester et al., 1990) came up with a seminal model, latent semantic indexing (LSI). They applied singular value decomposition (SVD) to summarize a document term matrix to its latent factors. (Landauer & Dutnais, 1997) improved the LSI model creating the latent semantic analysis model (LSA). They made the assumption that words with close meaning occur in similar documents. (Hofmann, 1999) extended the LSI model by incorporating a generalized Expectation Maximization algorithm with pLSI approach. The model could deal with polysemy and synonymy unlike LSA and LSI.

These pioneering models paved the way for (D. M. Blei et al., 2003) who came up with LDA. The main improvement in LDA was the inclusion of a probabilistic model at the document level, by assuming that documents are a mixture of topics. In pLSI there is no probabilistic model for documents in a corpus. This solved two main problems of the pLSI model (1) the number of variables increase linearly as the size of the corpus grows, this may lead to overfitting. (2) It is not generalizable outside of the training set.

The inclusion of a second probability component at the document level in LDA introduced the two-tiered model typical of the topic model framework. In this model, documents are presumed to be a mixture of topics and topics a combination of words. Each topic unique occurrences of words.
The introduction of mixture components in LDA led to problems in estimating the optimal model due to exponentially large potential topic values. This led to questions of what was the best way to compute topic models.

2.2.3 Computational Methods.

There are two main methods of computing inference for topic models: sampling methods (e.g., Gibbs Sampling) and variational inference. Gibbs Sampling was introduced by (Griffiths & Steyvers, 2004) and uses a Markov Chain Monte Carlo algorithm for inference of LDA. The algorithm estimates the Dirichlet priors in order to approximate the true posterior.

Another approach to estimating the posterior are variational inference methods that are an optimization problem. Variational methods hypothesize a parameterized group of distributions over the latent structure and then find the one that is closest to the posterior (D. M. Blei, 2012). (Blei et al., 2003) introduced the Expectation Maximization (EM) algorithm for variational inference. EM uses Kullback-Leibler (KL) divergence to best estimate the posterior as close as possible to the true posterior.

2.2.4 Extensions to LDA

Since it was introduced, LDA has been adapted and extended in several ways. LDA is explained by the statistical assumptions it makes about the corpus. An active area of research in topic modelling is how to modify these assumptions to discover more advanced structures in the texts (Blei, 2012).

2.2.5 The correlated topic model

One constraint of LDA is that it does not represent correlation between the latent topics (D. M. Blei & Lafferty, 2009). This constraint is because it uses the Dirichlet
distribution to model the topic proportions. Naturally, the occurrence of latent topics in most corpora will be correlated.

In the correlated topic model (CTM), (D. Blei & John, 2007) topics are modelled using the logistic normal distribution. This is a more flexible distribution that accounts for covariance pattern among the proportions. This gives a more natural representation of the topics where one topic may be correlated with another.

2.2.6 The dynamic topic model

LDA and CTM ignore the word order within the documents. They further assume that the order of documents within the corpus does not matter (D. M. Blei & Lafferty, 2006), this assumption is inappropriate. This is especially true when examining documents that span a long period.

The dynamic topic model (DTM) (Blei & Lafferty, 2006) models the evolution of topics in a serially catalogued corpus of documents. In DTM, the documents are divided by time, e.g., by month or year.

2.2.7 Author-topic model

(Michal Rosen-Zvi, Griffiths, Steyvers, & Smyth, 2012) extended LDA to include document information about the author. Every author is linked to a multinomial distribution over the topics while each topic is a distribution over the words. By modelling the interests of the authors, it can be established which topics the author writes about. Documents by different authors can be compared to establish which authors create similar work. Using LDA, the only way to examine the effect of the author was by manually examining how topics change according to the author.
2.2.8 Structural topic model

The structural topic model (Roberts et al., 2017) (STM) extends LDA by allowing users to include arbitrary metadata into the topic model. The goal of the STM is to discover topics and model their association to the document metadata. Hypothesis testing about these associations can be done using the model output. The model also introduces improvements to the model inference methods in order to render the model applicable to advanced modelling and evaluation.

2.3 Empirical Review

(D. Blei & John, 2007) applied the correlated topic model (CTM) to science articles from the JSTOR archive published from 1990 - 1999. The dataset comprised of 57M words. In most science topics, we can presume that there will be a high correlation within the hidden topics. They showed that it estimated the topics better than LDA, as measured by the accuracy of predictive distributions over the test set documents.

(Rosen-Zvi et al., 2012) applied the author-topic model on the NIPS data set consisting of papers from NIPS conferences. Researchers focused on learning algorithms and computational neuroscience contribute to NIPS. They illustrated the top 10 words most likely associated with a topic and the top 10 authors most probable to have written a word associated with the topic. They showed that for each topic, the top 10 most likely authors identified for each topic are popular for papers written on those topics.

(Roberts et al., 2014) used the structural topic model (STM) to analyze open-ended survey responses. Open-ended surveys are more difficult to analyze than closed surveys since human coding is not always applied. They showed that the model was
better than using human coding in a couple of ways. First, it permits the analyst to uncover topics from the data instead of guessing them. Second, the model allows analysts to study how frequency and content of topics changes with metadata that is associated with each respondent e.g., background demographic data. One advantage of using the model instead of human coders is that it allows an analyst to discover topics rather then speculate them. Secondly, it allows a scientist to discover the topics while studying how they prevalence and content changes with metadata particular to each respondent. They argued that the model can be used either at the exploratory stage of analysis or to make rational inferences about the effect of treatments on the content of open-ended survey responses. They showed that for a topic to be semantically interpretable, it had to have two qualities: (1) It has to be cohesive in that high probability words for a topic co-occur within the documents, and (2) exclusivity in that top words for a topic are unlikely to occur as top words in other topics. They applied structured topic model to analyze open-ended survey responses from the American National Election Survey (ANES). It had a sample of 2,223 respondents interviewed after the 2008 presidential election. Results differentiated between democrats and republicans. Democrats wrote more about the Iraq war than republicans did. They showed that the benefits of using the unsupervised STM model far outweighed the costs of human coding.

(Bhattacharya, Ganguly, Ghosh, Zafar, & Gummadi, 2014) developed an approach to discover topics that individual users of Twitter are interested in. They observed that Twitter users generally follow experts in various topics in order to gain information on those topics. They observed the users a user was following and then
identified topics of interest of those users. They concluded that if a user follows several experts on a particular topic, they are likely to have an interest in that topic. 

(Reich, Tingley, Luis, Roberts, & Stewart, 2015) applied structural topic modelling to analyze the massive quantities of text generated by students of Massive Open Online Courses (MOOCs). They showed how structural topic modelling can be used to find patterns with semantic meaning in unstructured text and identify how those patterns vary with respect to the document level metadata. In traditional learning environments, educators can read and process student generated text data in a timely fashion. However, for Massive Open Online Courses there is too much data that it would be overwhelming for the educators to read and analyze in reasonable time. They obtained data posted by students of MOOCs in course evaluation forms, discussion forums and pre-course surveys. They demonstrated how the structural topic model could assist instructors in understanding themes in student posts and show how the distribution of topics varies by important features of each document. They incorporated covariates such as age and gender to show the effect they have on the topics derived. They discovered that young MOOC students are more motivated by the association to the elite universities offering the courses while older students were more interested in career development.

(Lucas et al., 2015) analyzed how the procedures for processing, managing, translating and analyzing textual data differ across languages. They then applied the structural topic model on different religious documents and social media posts to compare how they differ between countries. They compare how different news agencies cover news related to China by analyzing stories from Xinhua and Agence France Presse (AFP). They also compared different Muslim clerics depending on whether they were
Jihadist or non-Jihadist. They showed differences topic proportions with regard to fighting and excommunication topics between the two.

(Sokolova et al., 2016) performed topic modelling and event identification from Twitter data. They worked on four datasets collected by the Umati project through Twitter’s streaming API: (1) The Gikomba Twitter data mainly covering a bombing incident in Nairobi’s Gikomba market. The dataset had 482 tweets. (2) The Mandera Twitter data that contained tweets on tribal clashes in Mandera town in Kenya. The data had 915 tweets in total. (3) The Makaburi dataset containing 20642 tweets. In those tweets, people were talking about the violent death of a controversial Muslim cleric, Sheikh Makaburi. (4) The Mpeketoni dataset containing 106348 tweets. In those Tweets, people discuss a terrorist attack that happened in Mpeketoni, a town in the coastal region of Kenya. They applied LDA for topic modelling after the initial steps of data pre-processing. They then analyzed the topics manually and by using topic coherence analysis. Topic coherence measures each topic by scoring it using the level of semantic similarity of words in a topic.

(Kuhn, 2017) used structural topic modelling, a machine learning technique, to explore aviation safety reporting data. This data has more than a million confidential reports outlining aviation safety incidents. He showed that structural topic modelling could identify topics contained with a vast amount of documents. Structural topic modelling could also estimate the impact of document level covariates such as time on the topic prevalence. This technique could be used to reveal hidden trends in the prevalence of topics in which natural meanings appear. He used the stm package for structural topic modelling in the R statistical software. He used the trade-off between semantic coherence
and exclusivity to select the number of topics to identify. Semantic coherence shows how often distinct words occur and bi-grams co-occur. In general, as the number of topics increases, the semantic coherence will decrease. Exclusivity of a topic measures if words that have a high probability of appearing in one topic conversely have a low probability of appearing in other topics. In general, as the number of topics increases, the exclusivity of the model increases. He showed that the most prominent topics identified involve air traffic control and takeoff and landing stages of the flight. The identified topics also revealed the importance of the fuel pump, fuel tank and landing gear. He made use of document level covariates to show that issues fire and smoke are most likely to occur in cargo and passenger planes. He also particularly focussed on data from the San Francisco International Airport because of the vast number of records linked to it. He showed that passenger planes are more likely to report issues related to approach to the airport. Private airplanes were more likely to report issues related to the taxi topic. The application of structural topic modelling also showed the importance of the Quiet Bridge Visual and the Tip Toe Visual landing paths to the airport. Many air traffic operators and analysts are already familiar to issues related to these landing paths.

(Schwemmer & Jungkunz, 2019) investigated the representation of indigenous groups and women in TED talks using structural topic modelling. TED talks are free talk videos posted online covering the topics of technology, entertainment and design. They made use of an image recognition algorithm to identify the gender and ethnicity of a speaker and compiled a dataset from TED talks and youtube comments. They observed that more than half of the talks were presented by white male speakers while the share of women speakers has increased but the share of non-white speakers remains relatively low. They
observed that insufficient representation of both groups can create an inaccurate impression of science and other topics discussed in the talks. They also analyzed the response of the online audience to the representation of these marginalized groups so as to gain insights on the public perception to these groups and topics they are interested in. They applied structural topic modelling on all TED talks transcripts to identify talks in which ethnic minorities and women issues were addressed. Sentiment analysis of YouTube comments suggested that public sentiment was positive for non-white speakers but negative for talks about inequalities such as racism and violence against women. Talks by women also received negative sentiments than those given by men. Using the structural topic model, the authors extracted 30 topics from the dataset. They identified one of the topics to be strongly related to the representation of indigenous groups and women. They labelled the topic ‘inequality’ as the talks predominantly addressed inequality in the treatment of the marginalized groups. This topic accounted to about 3% of all the transcripts analyzed. The most important terms identified by the topic indicated by the frex metric were identified. They showed the topic captured content about gender and sexuality. References to ethnicity, violence and abuse of power were also identified. They also analyzed how the topic is correlated to other topics as it is allowed in the structural topic model. The topic clusters indicated the topic was correlated to other topics about family, school and children but also to a lesser extent with topics about war, politics, terror and law. The authors also included gender and ethnicity as covariates in the estimation of the topic model. They showed that women were less likely to give talks about computers and technology and that white male speakers dominated the environment topic. They concluded that the inequalities for women marginalized ethnic
groups can be addressed by increasing the share of talks by these groups consistently. Furthermore, they are likely to talk about the inequities they experience.

2.4 Conceptual Framework

As guided by the literature review, the following conceptual framework was developed. It shows how the Twitter corpus constructed from a trending topic will be used to derive topics users are discussing. Topic models are unsupervised algorithms, therefore, the variables will be inferred from the corpus. The conceptual framework is illustrated in the figure below.
Figure 2.1: Conceptual Framework

- **Twitter Text – Message posted on Twitter**
- **Date Created – Date the message was posted**
- **Source – Twitter application used to post message**
- **K – Number of Topics**

**Dependent Variable**: Derived Topics
2.5 Operationalization of Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Indicators</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Twitter Text</td>
<td>Text posted on Twitter</td>
<td>Text</td>
</tr>
<tr>
<td>Date Created</td>
<td>Date post was created</td>
<td>Date/Time</td>
</tr>
<tr>
<td>Source</td>
<td>Application used to create the post</td>
<td>Twitter Web App</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Twitter for Android</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Twitter for iPhone</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TweetDeck</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hootsuite .Inc</td>
</tr>
<tr>
<td>K- Number of topics</td>
<td>Number of topics to be estimated</td>
<td>5-50</td>
</tr>
</tbody>
</table>

Source: Author (2019)

Table 1: Operationalization of Variables
CHAPTER THREE
RESEARCH METHODOLOGY

3.1 Introduction

This chapter will examine the research methods that will be utilized in order to achieve the objectives of the study. This chapter also covers the research design to be utilized, data preprocessing methods, data analysis methods and model evaluation.

3.2 Research Design

Research design is the general framework within which the research will be carried out. It involves identifying the source and method of obtaining the relevant data and methods for analyzing the data. This study will adopt the data mining methodology that defines procedures and rules for managing a data mining process. This methodology will ensure that the data mining process will lead to a robust model that will solve the problem at hand.

Data for this research will be obtained from the Twitter search API. The API returns Tweets matching the user-defined query. Not all tweets matching a specified query are made available via the Search API. The search will include only Tweets in English and will exempt retweets.

![Figure 3.1 : Research Design](image-url)
Data collected via the Twitter API will be stored in a MySQL database. Common text preprocessing methods will then be applied to the datasets. Once all preprocessing is completed, the remaining terms will be converted into a document-term matrix (DTM). A DTM is a data structure where a document is represented by a row on the matrix and each unique word is represented by a column. It uses the “bag-of-words” approach where word order is discarded but the number of times each word occurs is recorded. Automated content techniques such as LDA or STM will then be applied to this matrix to learn the topics.

3.3 Data Analysis

The data will be analyzed using the R language. R has many open source implementations for topic modelling. In this case I will use the stm R package (Roberts et al., 2017). The choice of this package is informed by the fact that it allows modelling of document metadata and provides a statistical-based framework to enable hypothesis testing.

The stm is a package that allows researchers to estimate topic models with document level metadata. If the researcher does not include the metadata, it reduces to a fast implementation of the correlated topic model (CTM).

To use the package, first the data is read and prepared for analysis. Then a structural topic model is approximated. The package provides functions for evaluation, understanding and visualization of the results, as shown in the diagram below.
3.4 Data preprocessing

Twitter data is considered more challenging than other social media data due to character limit, misspellings and slang (Eisenstein, 2013). In this study, Twitter specific text preprocessing will be performed. This will include removal of Twitter-specific features like hashtags, hyperlinks, emoticons, user mentions, slang and acronyms.
Other common text pre-processing procedures that will be done include stop word removal, stemming, lemmatization, Tokenization and converting to lowercase.

3.4.1 Stopword Removal

This involves removing recurring words such as “and” and “the” to aid in model performance and interpretation. These words do not contribute to the meaning of the document. The most common way to remove them is to use a fixed list. Such lists are available in many languages and typically take care of most stopwords.

3.4.2 Stemming and Lemmatization

Stemming gets rid of the letters added to conjugated verbs and nouns leaving just the base form of the word. Lemmatization is an advanced algorithm that returns the *lemma*, or canonical form of a word.

3.4.3 Tokenization

Tokenization is the process of separating a string of text into its constituent words. For English, whitespace and punctuation are usually used to detect word boundaries. Tokenization is the first step before creating a document term matrix.

3.4.4 Converting to Lowercase

For a particular concept there can be many character strings representing it depending on the case. For example, “Topic”, “TOPIC” and “topic” refer to the same underlying concept. Converting to lowercase will convert them to the same word for purposes of topic modelling.

3.4 Choosing Parameters

After data pre-processing and creation of the document-term matrix. There are a number of parameters the researcher needs to choose before running the model. First is
the number of topics (K) which indicates the number of topics that should be identified by the model (Jacobi, Van Atteveldt, & Welbers, 2016). There is no default value for this parameter. The aim is to represent the documents with fewer than the actual number of topics present but with as little loss to relevant information as possible.

Second is a hyperparameter, alpha, which affects the number of topics to be identified within the documents. A widely used default is 50 divided by K.

3.5 Model Evaluation

The results of the topic model will be evaluated using held-out likelihood/perplexity and manual analysis of each topic. Manual analysis will be done by examining each topic and the top words closely. Perplexity will also be used to evaluate the topic model. Perplexity can be considered as a measurement of how well a probability distribution predicts a sample. If a topic model has low perplexity, then it is considered more generalized, compared to the one that has high perplexity.
CHAPTER FOUR
DATA ANALYSIS, FINDINGS AND DISCUSSION

4.1 Introduction

This chapter presents analysis of the data collected and findings of the study. Results for research the objectives will be discussed in this chapter. This study will use data collected via Twitter API and use Structural Topic Models to extract relevant topics discussed by users utilizing particular hashtags.

4.2 Descriptive Statistics

This study focuses on Twitter messages related to the useR 2019 conference that took place between July 10 and July 15 in Toulouse France. This is an annual gathering of R community users. It involves talks by leading experts in the R community and conference sponsors. The Twitter API provides several ways to download Tweets. The search API was used which allows retrieval of tweets posted in the past seven days. Tweets retweeted by users were not collected as this would potentially result in many duplicate Tweets collected. The data collected contains 23,000 tweets posted by 5031 users. Tweets were collected using several hashtags related to the conference to bypass Twitter rate limit. The graph below shows the frequency of tweets containing #user2019, #Rladies, #LatinR, #AfricaR, #Rshiny, #tidyr, #dplyr hashtags between July 07 and July 15.
The figure shows most tweets were posted between July 10 and July 12 when the conference was starting and the frequency of tweets decreased towards the end of the conference.

4.3 Research Findings

4.3.1 Objective one Results

The first objective of this thesis was to determine the text preprocessing techniques that are most appropriate to apply to Twitter messages. Most of the tweets collected were in the English language as shown below. However, there were tweets posted in other languages like Spanish and French. Only tweets in English were retained since mixing different languages would be problematic in the model interpretation. The final dataset after filtering the languages contained 21,773 tweets.
Figure 4.2: Language used for collected tweets

Several text-preprocessing procedures were applied to the tweets. Some were procedures specific to tweets text and others were general text preprocessing procedures. The procedures specific to tweets text included removing hashtags, URLs and other non-ascii characters such as emoji’s. A dummy variable was also added representing tweets posted in the first days and those posted in the last days. This variable divided the dataset into two and will be used as metadata to be used in the model representing content for the different days of the conference. The other more general procedures included converting to lowercase, removing punctuation, removing stopwords, removing numbers and stemming. This was done using the textProcessor function from the stm package as shown below.
Figure 4.3: Text preprocessing procedures

The output of this function was then fed to the prepDocuments function also from the stm package. It performs several corpus manipulations such as removing words and renumbering word indices as shown below. The corpus is now ready to be modeled using the stm package to estimate topics.

Figure 4.4: Creating a corpus for modeling in stm

4.3.2 Objective two Results

The second objective of this thesis was to apply a topic model to Twitter trending topics. The stm R package was selected due to the many functionalities it offers for text preprocessing, estimating and evaluating topic models. It also has an added advantage compared to other packages because document specific metadata can be incorporated when estimating topic models. The metadata can affect the frequency with which a topic appears or the words which appear in a particular topic.

The stm package contains several methods for determining the appropriate number of topics for a particular dataset. (Roberts et al., 2017) recommend using the Spectral initialization in the stm function as this has been shown to produce the most consistent results. There is no “correct” answer to the number of topics for a particular corpus (Grimmer & Stewart, 2013). The stm package provides the searchK function for
comparing different number of topics and selecting the best using automated tests. For this corpus 5, 10, 15 and 20 topics were tested and it was determined that 20 was the most appropriate number of topics.

Figure 4.5: Diagnostic values by number of topics showing the values peak at 20 topics

The figure above shows the diagnostic results for the different number of topics tested. The results show that the held-out likelihood is highest at 20 topics and the residuals are lowest at around 20 topics.

Semantic coherence is highest when most probable words for a given topic appear together (Stevens, Kegelmeyer, Andrzejewski, & Buttler, 2012). It is high when there are few topics dominated by most common words. From these observations, we can conclude that the optimal number of topics for this corpus is 20 topics.
With the number of topics to be estimated selected, the next step is to fit the structural topic model. There are four ways to initialize the model namely, Latent Dirichlet Allocation (LDA), Random, Custom and Spectral. This study will use Spectral initialization as it has been shown to provide the most consistent results. Initialization calculates the gram matrix and finds the anchor words. Anchor words are words unique to each topic and therefore identify that topic. Once the anchor words are found, the model completes the initialization. The model then performs several iterations to derive the topics until it converges.

```
Beginning Spectral Initialization
Calculating the gram matrix...
Finding anchor words...
....................
Recovering initialization...
.............................................................
.
Initialization complete.
..............................................................................
......................
Completed E-Step (11 seconds).
Completed M-Step.
Completing Iteration 1 (approx. per word bound = -7.772)
..............................................................................
......................
Completed E-Step (9 seconds).
Completed M-Step.
Completing Iteration 2 (approx. per word bound = -7.401, relative change = 4.775e-02)
..............................................................................
.............
Completed E-Step (8 seconds).
Completed M-Step.
Completing Iteration 3 (approx. per word bound = -7.159, relative change = 3.262e-02)
..............................................................................
.............
Completed E-Step (10 seconds).
Completed M-Step.
Completing Iteration 4 (approx. per word bound = -6.945, relative change = 2.992e-02)
..............................................................................
.............
Completed E-Step (8 seconds).
Completed M-Step.
```
Below are the results of fitting the model with 20 topics. The most probable words for each topic have been plotted. The topics can then be labeled manually according to the words that appear in them. For example topic one talks about the keynote of the conference and the community and topic two talks about the mood of the people in the conference.
From this visualization, we can see that different words have different probabilities for each topic and each topic consists of different words.

The words in each topic can also be visualized as word cloud as shown below for topic one.
The `stm` package also has the `labelTopics` function which generates the top words per topics using different metrics like, highest probability, FREX and lift scores. The probability score outputs the highest probability words for a topic. The FREX score weighs words by how exclusive they are to a topic by their overall frequency. The lift score weighs words by dividing with their frequency in other topics, therefore giving more weight to words that are less frequent in other topics. Score divides the log frequency of a word in the topic with the log frequency of the word in other topics. The output of the function for the twenty topics is shown below.

**Topic 1 Top Words:**

- **Highest Prob:** communiti, make, come, peopl, juliesquid, keynot, open
- **FREX:** seek, forc, divers, mozilla, ncea, artwork, teamwork
- **Lift:** divers, megaphon, sticki, teamwork, --formal, -stress, abichat
Score: juliesquid, allisonhorst, keynot, communiti, seek, peopl, feedback

Topic 2 Top Words:
Highest Prob: today, will, take, month, one, confer, book
FREX: month, win, advantag, trial, women, sign-, end
Lift: -kms, aithm, almost, aperio, aussi, brycem, canal
Score: sign-, crcmathstat, advantag, month, trial, crcpress, today

Topic 3 Top Words:
Highest Prob: ggplot, make, nice, happi, sheet, differ, error
FREX: sheet, cheat, street, rmarkdown, factorccharact, factorcinteg
Lift: -time, accident, accidentalart, aesx, align, amd, ann
Score: sheet, factorccharact, factorcinteg, ggplot, street, cheat, nice

Topic 4 Top Words:
Highest Prob: present, model, much, paper, system, room, method
FREX: benchmark, rmd, attempt, life, render, demystifi, orient
Lift: lavaan, adviceresourcessuggest, assist, attempt, bake, binomi
Score: present, model, benchmark, orient, malign, demystifi, system

Topic 5 Top Words:
Highest Prob: user, two, here, chang, year, dplyr, time
FREX: interview, studi, market, glue, fan, warn, secur
Lift: -minut, abid, aetherczar, agesex, akur, amarantafocardi, american
Score: user, here, died---wool, interview, glue, bang, chang

Topic 6 Top Words:
Highest Prob: talk, userconf, great, look, shini, session, app
FREX: forward, contribut, thinkrfr, product, lightn, java, golem
Lift: bank, causaldisco, credenti, cutest, golem, istevess, mirai
Score: talk, userconf, shini, app, great, session, look

Topic 7 Top Words:
Highest Prob: new, packag, cran, version, dont, document, featur
FREX: version, juliasilg, control, patch, forum, dbplot, movement
Lift: -hous, -memori, adapt, ampersand, ampl, asyk, auditorium
Score: new, version, cran, document, releas, dont, initi

Topic 8 Top Words:
Highest Prob: use, now, like, post, function, made, can
FREX: keep, home, r-blogger, stuck, bad, convers, fomo
Lift: icit, -caus, acabei, accent, aden-bui, admit, ador
Score: now, post, like, fomo, r-blogger, convers, keep

Topic 9 Top Words:
Highest Prob: first, share, interest, will, ill, project, also
FREX: first, colinfay, minut, feast, fabl, preview, octav
Lift: analyticsinm, fabl, preview, -point, accumul, addin, adrianrafteri
Score: first, ill, share, project, feast, interest, minut

Topic 10 Top Words:
Highest Prob: learn, use, machin, updat, need, analysi, tutori
FREX: python, languag, bring, web, kubernet, hoai, interpret
Lift: -hour, alfr, analyxcompani, analyzefmri, apachearrow, asymmetri,
Score: learn, machin, updat, cran, supervis, unsupervis, tutori
Topic 11 Top Words:

Highest Prob: data, scienc, just, love, time, turn, cool
FREX: turn, cool, raw, disciplin, inclus, non-elitist, vector
Lift: awe, bandsmultipl, breweri, carb, chatter, controversi, ctrl
Score: data, scienc, disciplin, love, non-elitist, raw, just

Topic 12 Top Words:

Highest Prob: data, plot, network, column, function, power, neural
FREX: column, azur, row, infer, storag, level, lake
Lift: accur, agegraph, atp, baffl, bah, bottom, consecut
Score: lake, column, data, gen, neural, storag, azur

Topic 13 Top Words:

Highest Prob: can, see, day, free, live, program, hadleywickham
FREX: day, live, ebook, ciso, hexagon, object-ori, neat
Lift: object-ori, africar, approv, athosdamiani, atsushi, best-sel, blow
Score: day, ebook, see, free, ciso, hexagon, can

Topic 14 Top Words:

Highest Prob: data, use, packag, visual, graphic, creat, base
FREX: graphic, scatterplot, focus, grammar, wilkinson, trust, eas
Lift: -dev, aapelinevala, acid, adalthousephd, aedinculhan, ann-mari
Score: data, scatterplot, graphic, wilkinson, eas, multi-lay, stun

Topic 15 Top Words:

Highest Prob: start, want, packag, know, find, tutori, miss
FREX: avail, catch, usethi, themlet, local, formula, mathemat
Lift: asnumerica, baker, bryan, casnumerica, ceram, datavisfriend
Score: themlet, want, find, tutori, materi, start, miss

Topic 16 Top Words:

Highest Prob: thank, slide, copyright, today, amp, confer, everyon
FREX: everyon, exercis, livestream, franc, obvi, reexam, gorgeous
Lift: aaronhamilton, accentur, accommod, apolog, basin
Score: slide, thank, everyon, copyright, gorgeous, toulous, livestream

Topic 17 Top Words:

Highest Prob: get, work, way, anyon, use, ive, lot
FREX: jcheng, problem, recommend, actual, behind, shinymeta, relat
Lift: experiment, adminlt, allan, await, blast, boulangeriebar, bust
Score: work, jcheng, get, lot, survey, reproduc, ive

Topic 18 Top Words:

Highest Prob: workshop, run, amp, better, help, juli, scientist
FREX: report, yesterday, summer, answer, school, particip, environment
Lift: amandamioottogu, asap, birkbeck, career, cdt, concret, concurr
Score: workshop, juli, edzerpebesma, run, better, scientist, answer

Topic 19 Top Words:

Highest Prob: join, communiti, twitter, follow, week, meet, list
FREX: stay, romainfrancoi, revodavid, palett, stori, achimzeilei, deceiv
Lift: aaronjfish, abresl, achim, adag, alicedata, ami, amsterdam
Score: deceiv, illus, join, stay, twitter, revodavid, network

Topic 20 Top Words:
So as to find the most representative documents for each topic, we can use the `findThoughts` function in the `stm` package. This will help us get a better view of the actual documents with high topical content. The output of the `findThoughts` function produces output as shown below.

```
> findThoughts(rstats_stm, texts = rstats2$text)
```

Topic 1:

One one hand, I'm unsure if this debugging line in an error: "Underfull \vbox (badness 10000) detected at line 119" On the other, I'm amazed that the badness is over 9000.

@filipstachura summarized at what we learned about large scale R Shiny deployments. Shiny is indispensable for fast prototyping and reporting, but now it is also ready for large production applications.

I didn't know you can use to make conference posters! Cool and useful! Would love to see some examples!

Topic 2:

Grand prize for @UseR2019_Conf. An autographed copy of Advanced R by @hadleywickham. To be eligible to win, you must be an attendee, retweet this and follow me. Good luck to all! I will announce the winner at the end of the conference.

@CRC_MathStats @CRCPress
Grand prize for @UseR2019_Conf. An autographed copy of Advanced R by @hadleywickham. To be eligible to win, you must be an attendee, retweet this and follow me. Good luck to all! I will announce the winner at the end of the conference.

@CRC_MathStats @CRCPress function: head - list the first six rows

factory. instrospection may suggest that some separation of "church" and "state" is needed

Just x-raying a bunny in (in reality, this is just a visualization of the number of bounding volume hierarchy intersects in , and you can see by the random fluctuations that its structure is totally random right now-going to implement some optimizations soon)

The structural topic model allows for the inclusion of document level metadata. This study allowed for the inclusion of this feature by including a dummy variable indicating the period a tweet was posted. The dummy variable “one” indicates that the tweet was posted in the first days of the conference and the dummy variable “two” indicates the tweet was posted during the final days of the conference. The figure below shows the difference in perspectives introduced by the covariates for topic 15. This allows for the modelling of the temporal perspective in the data.
Figure 4.8: Plot of different perspectives for topic 15.

The structured topic model allows the topics to be correlated. The figure below shows how the topics are correlated in the model. The width of the links represents the strength
of the correlation and the size of the nodes represents the topic proportions.

Figure 4.9: Plot of correlation between the topics.

4.3.3 Objective three Results

The third objective of this thesis was to test and validate the effectiveness of the topic model in identifying the relevant topics from Twitter trending topics. Evaluation of the effectiveness of topic models is inherently difficult due to the complexity of the underlying model with multiple solutions space (Wallach & Mimno, 2009). However, a good evaluation metric is the probability of held-out documents given a previously trained model.
The stm package provides two functions to estimate the held-out likelihood of a topic. The `make.heldout` removes some words from the documents from which a topic model will be estimated. The `eval.heldout` function evaluates the held-out likelihood probability for the words that were removed on the model run on the heldout documents.

The held-out likelihood measures how well the model fits a corpus of unseen documents (Chang, Gerrish, Wang, & Blei, 2009). The higher the held-out likelihood the better the model captures patterns for the corpus. This is similar to the tests carried out when selecting the best value for K in order to estimate the best model.

The results of the evaluation as shown below is consistent with the searchK for 20 topics that was run earlier. Because we are only running the stm topic model, we can only evaluate according to the number of topics. The highest held-out likelihood was observed with 20 topics as shown below.

```
> eval_heldout <- eval.heldout(stm_eval, heldout$missing)
> eval_heldout$expected.heldout
[1] 6.781996
```

**Figure 4.10: Results for held-out likelihood test.**

Word intrusion is another metric that can be used to evaluate topic models (Chang et al., 2009). This is used to measure the coherence of a topic by determining which word is out of place in a topic. From our model, we can observe that all topics have a common theme and no word is out of place.

**4.4 Discussion of Results**

The results of this study are consistent with other empirical literature relating to structural topic modelling. (Schwemmer & Fischbach, 2019) applied structural topic modelling to understand why people used Twitter in the aftermath of a terror attack. The
general approach from data collection, text preprocessing, model selection and evaluation is consistent with the methods applied throughout this study.

In this study, structural topic modelling was used to estimate a topic model from a Twitter corpus. A data driven method was used to determine the optimal number of topics to estimate. Twenty topics were estimated from the corpus and visualized using different charts. The model was then evaluated using the held-out likelihood method and word intrusion, the results showed that the model performed well in estimating the topics.

It has been shown that topic models can be a useful tool in summarizing large text corpora. The models can be used in exploratory data analysis and in predicting topics for unseen documents.

4.5 Summary

In this chapter, data was obtained from Twitter using the Twitter search API. The appropriate text preprocessing procedures were identified and a Document Tem Matrix (DTM) object was created for use in estimating the topics. The findings of this study show that the structural topic model extracts useful topics from Twitter data. The topics consist of words which co-occur from the corpus which are grouped together to form a topic. The topics can then be labelled manually according to the words appearing in the topic. This is useful in summarizing discussions of users showing themes of common interest among the users. This information can be useful to various stakeholders as discussed in significance of the study.
CHAPTER FIVE

CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter summarizes the findings of the study, discusses conclusions and recommendations to advance this work.

5.2 Conclusions

In this study topic modelling has been shown to be effective in exploring a large text corpus. The results can be useful in exploratory data analysis and in text classification. This study has explored how data can be obtained, prepared and modelled to produce a useful model.

In order to obtain accurate results, text preprocessing procedures are an important step in the analysis process. These procedures such as converting to lowercase, removing stopwords and punctuation and stemming ensure that clean data is fed into the model.

Before the model can be run, hyperparameters have to be set. Hyperparameters for stm include number of topics (K) and the initialization type. The optimal number of topics for a particular dataset can be determined by running the searchK function contained in the stm package. The initialization hyperparameter is used to find words that belong to one topic and therefore identify that topic. The structural topic model the initialization types include Spectral, LDA, Random and Custom. Spectral initialization is recommended in most cases as it produces the most consistent results (Roberts et al., 2017).
Visualizations play a critical role in interpreting and evaluating topic models. Using bar charts, we can plot the most frequent words in each topic and their probability of appearing in the topic. Using the charts infringing words for a topic can be spotted. Word clouds can also be used to visualize all the words for a topic and their frequency of occurrence. The more frequent words are plotted in a bolder font than the less frequent ones.

Evaluation of topic models is not a trivial task due to the numerous possibilities when estimating the model. However, heldout likelihood/perplexity is a well-established method of evaluating topic models. It is used to measure how well a model fits to unseen documents. Word intrusion can also be used in model evaluation. This is done by identifying words that are out of place for a particular topic.

5.3 Contributions of the study

This study has made significant contributions to the field of topic modelling. In the past topic modelling algorithms have not allowed for the inclusion of document level metadata when running the model. This study utilized the structural topic model which allows for the inclusion of the metadata. Document level metadata in the form of dummy variables representing the period a tweet was posted. The assumption is that people will post different topics on different days of the conference. The study has shown how by incorporating the document level covariates, the topics can be plotted to show the temporal aspect. A distinction between the tweets posted in the first days was shown with the tweets posted during the final days by plotting the topics and including this perspective. This aspect can be extended to other studies with other metadata to show a distinction between the documents. This is a major distinction in the structural topic
model compared to other topic modelling algorithms. By incorporating this metadata, the
model performance is improved.

This study also shows how researchers can utilize the R programming language to
perform end-to-end data analysis. The R language has several programs known as
packages that researchers can utilize in all stages of data analysis. These include packages
for data collection, data cleaning, modelling, evaluation and visualization. This integrated
environment results in greater productivity as it eliminates the need to switch to different
tools.

5.4 Recommendations for Future Research

This study was not without limitations. The data, which also determines the
results, was obtained from Twitter. Twitter imposes limits by only exposing a subset of
tweets searched using a particular hashtag. This means that all tweets relating to the event
could not be captured. Therefore the results do not capture all individual posts about the
conference.

This study recommends that more research should be done in the field of topic
modeling, as it is a relatively new and continually evolving field. Researchers can use
data from other social networks, blogs, books or even reports.

Researchers can also examine how model performance is affected by the amount
of data used in modelling. Topic models are designed to handle large quantities of data.
The dataset used in this study is relatively small. Future research can incorporate bigger
datasets to investigate the effect on model performance and results.
Future research can also compare different topic modelling algorithms performance on different text datasets on various metrics in order to determine the best algorithms to apply in different situations.
REFERENCES

https://doi.org/10.1504/IJWBC.2013.051298

https://doi.org/10.1145/2645710.2645765

https://doi.org/10.1214/07-AOAS114


https://doi.org/10.1162/jmlr.2003.3.4-5.993


https://doi.org/10.1111/ajps.12103


https://doi.org/10.1080/2474736X.2019.1646102


APPENDICES

Appendix I: Budget

<table>
<thead>
<tr>
<th>Budget Items</th>
<th>Cost (KES)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal - Printing, binding and internet costs</td>
<td>30,000</td>
</tr>
<tr>
<td>Data collection – Internet cost</td>
<td>15,000</td>
</tr>
<tr>
<td>Data analysis and reporting a) Printing and stationery costs</td>
<td>20,000</td>
</tr>
<tr>
<td>b) Hard cover Binding</td>
<td>15,000</td>
</tr>
<tr>
<td>Transport to Campus, airtime costs</td>
<td>20,000</td>
</tr>
<tr>
<td>Miscellaneous</td>
<td>50,000</td>
</tr>
<tr>
<td><strong>TOTAL BUDGET</strong></td>
<td><strong>KES 150,000</strong></td>
</tr>
</tbody>
</table>
### Appendix II: Work Plan

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposal Defense</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Collection and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preparation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Analysis and Report Writing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Defense and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Final Report Submission</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix III: Sample Source Code for the Structural Topic Model

# Import packages
library(lubridate)
library(ggplot2)
library(dplyr)
library(readr)
library(tidyr)
library(tidytext)
library(ggplot2)
library(stringr)
library(textclean)
library(topicmodels)
library(stm)
library(drlib)

# Remove unwanted characters
remove_reg <- "&|<|>"

#my_stop <- tibble(word = c("data", "@user2019conf", "de", "rt"))

# Read the dataset
rstats3 <- readRDS("~/R 2019/Research Project/rstats3.rds")

# Remove unwanted columns
rstats <- rstats3 %>% select(created_at, user_id, screen_name, text, source, lang, is_retweet)

# Add a day column
rstats <- rstats %>% mutate(day = mday(created_at))

# Add a document dummy variable
rstats <- rstats %>%
  mutate(rating = ifelse(day <= 9, "one", "two"))
# Remove retweets
rstats <- rstats %>% filter(!str_detect(text, "^RT"))

# Remove non english
rstats <- rstats %>% filter(lang == "en")

# Remove Retweets
rstats <- rstats %>% filter(is_retweet == "FALSE")

# Remove HashTags
rstats$text <- replace_hash(rstats$text)

# Remove URLs
rstats$text <- replace_url(rstats$text)

# Replace non ascii
rstats$text <- replace_non_ascii(rstats$text)

# Estimate stm
processed <- textProcessor(rstats$text, metadata = rstats)
out <- prepDocuments(processed$documents, processed$vocab, processed$meta)

rstats_stm <- stm(documents = out$documents, vocab = out$vocab, content = out$meta$rating, K = 20,
                  init.type = "Spectral", data = out$meta)

# Tidy STM
rstats_topicsstm <- tidy(rstats_stm, matrix = "beta")

top_termsstm <- rstats_topicsstm %>%
group_by(topic) %>%
top_n(5, beta) %>%
ungroup() %>%
arrange(topic, -beta)

#plot STM
```r
top_termsstm %>%
  mutate(term = reorder(term, beta)) %>%
ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered() +
  labs(title = "Highest word probabilities for 20 topics")

library(rtweet)

rstats3 %>% ts_plot("3 hours") +
  ggplot2::theme_minimal() +
  ggplot2::theme(plot.title = ggplot2::element_text(face = "bold")) +
  ggplot2::labs(
    x = NULL, y = NULL,
    title = "Frequency of #useR2019 statuses for collected Tweets",
    subtitle = "Twitter status (tweet) counts aggregated using three-hour intervals",
    caption = "\nSource: Data collected from Twitter's REST API via rtweet"
  )

rstats3 %>% ggplot(aes(x = rstats3$lang)) +
  geom_bar() +
  ggplot2::theme_minimal() +
  ggplot2::theme(plot.title = ggplot2::element_text(face = "bold")) +
  ggplot2::labs(
    x = NULL, y = NULL,
    title = "Language used for collected Tweets"
  )
```
rstats %>% ggplot(aes(x = rstats$rating)) +
  geom_bar()

heldout <- make.heldout(out$documents, out$vocab)
documents <- heldout$documents
vocab <- heldout$vocab
meta <- out$meta

# model evaluation
stm_eval <- stm(documents, vocab, K = 20, content = out$meta$rating,
    init.type = "Spectral",
    data = meta, max.em.its = 50)

eval_heldout <- eval.heldout(stm_eval, heldout$missing)