

https://dspace.org/

Mining tweets on sexual violence in South Africa

Imuede, Jude

2020
University of the Witwatersrand, Johannesburg
https://hdl.handle.net/10539/30194
Imuede, Jude. (2020). Mining tweets on sexual violence in South Africa. [Master's dissertation, University of the Witwatersrand, Johannesburg]. WIReDSpace. https://hdl.handle.net/10539/30194

Downloaded from a DSpace repository. https://dspace.org/

Mining Tweets on Sexual Violence in South Africa

By

Jude Imuede

(1692213)



WITS
UNIVERSITY

School of Computer Science and Applied Mathematics
University of the Witwatersrand

A dissertation submitted to the Faculty of Science in fulfillment of the requirements for the degree of *Master of Science (M.Sc)* in Computer Science

February 7 2020

Jude Imuede. 2020.
Mining Tweets on Sexual Violence in South Africa.

Copyright © University of the Witwatersrand, Johannesburg, South Africa.

All rights reserved. Not a piece of this dissertation is retrievable, transmitted, or duplicated through any storage means possible. This includes, but is certainly not limited to, photocopy, photograph, or magnetic, without the copyright holder's prior consent and written permission.

supervisors:
Dr. Pravesh Ranchod, University of the Witwatersrand
Dr. Mpho Raborife, University of Johannesburg

Declaration of Authorship

I declare that the content of this dissertation and the work presented therein are mine.

To this end, I establish the following.

- The Master of Science in Computer Science degree is being submitted to the University of the Witwatersrand, Johannesburg. The content therein has not been submitted to any other degree-awarding institution.
- The research, from conception to its developmental phases through to completion, is my work, except for quoted words from others.
- I have acknowledged all sources of reference. Otherwise, all views are mine.



Jude Imuede

February 7 2020

“By all means possible, be found in a place that does not compromise your security.”

—Jude Imuede

“Prediction is difficult, especially of the future.”

—Neils Bohr

“Everyone is entitled to their opinion about the things they read (or watch, or listen to, or taste, or whatever). They are also entitled to express them online.”

—John Scalzi

“The ability to simplify means to eliminate the unnecessary so that the necessary may speak.”

—Hans Hofinann

“If the ax is dull and its edge unsharpened, more strength is needed, but skill will bring success.”

—Ecclesiastes 10:10

“Education is a great engine for personal development.”

—Nelson Mandela

ABSTRACT

The "wisdom of the crowds" emerges from mining tweets incited by the behaviour of the masses. South Africans have resorted to venting their frustrations regarding the state of Gender-based violence (GBV) on social media. This is done in a bid to seek justice and raise awareness of the issue at large. Sexual violence is a kind of GBV that is widespread in South Africa, and Twitter is that popular social media and micro-blogging service where these frustrations are mostly reported as tweets. An effective analysis of Twitter data on sexual violence can expose unknown information and provide insights leading to better mitigating strategies. Within this context, this research investigates the disparity in reported cases of sexual violence in terms of gender representation and sentiment expression based on geolocation. It is envisaged that this information can be used in the form of an interactive data visualisation tool that policy analysts can use to investigate counter-preventive measures. Moreover, law enforcement agencies can better allocate their resources to help mitigate this problem. To do this, we achieved the development of a web-based, AI-driven application that automatically reveals the gender and sentiments in a tweet document. The system is built using the Estimator framework, a TensorFlow high-level-API which simplifies machine learning programming and model development. The process is initiated by the `Indexer` which is a server-side `Node.js` application that runs persistently and enables streaming of Twitter data for gender prediction, sentiment analysis and visualisation of GBV in South Africa.

KEYWORDS: Twitter API, Tweet, Gender prediction, Sentiment analysis, Sexual violence, TensorFlow, South Africa.

DEDICATION

To the soul of my resting MOTHER

Your love and support are eternally refreshing.

ACKNOWLEDGEMENTS

My studying experience at WITS University has truly been amazing. From day one of my M.Sc Programme in Computer Science, I have being beyond words, and even more excited as I soon discovered its state-of-the-art facilities in digital libraries and motivation for excellence. Indeed I was completely fulfilled and my expectations overreached. For this, I would like to specially acknowledge the many people who contributed in their different ways that have shaped this dissertation. Even more so, to the individuals who read and commented on earlier drafts of various portions of this research, or helped me work through the arguments in some way. In no order of preference, these include:

- Dr. Pravesh Ranchod, my principal supervisor, for your constructive feedback. My co-supervisor, Dr. Mpho Raborife, for your patience and unrelenting support. I am grateful for the opportunity to have been under your insightful model of supervision.
- Kgomotso Monyepote, Leeshia Roopnarian and Mpumi Mnaqapu, thank you for your immense support and your willingness to direct me and grant me access to relevant resources.
- Marike Kluyts, your sense of professionalism in mentoring is truly inspiring. I was happy to see our first joke exchange. Thank you in ways that I cannot explain.
- My friends: George Obaido, Patrick Iroanya, Michael Ayawei, Keith Benwalson, Douglas Okunubi, God'sgift Uzor, Kudakwashe Kahari, Ayomide Onasanya, Mandisa Mhlanga, and Tinozivashe Sibanda. I am glad our paths crossed in life.
- My treasure of a wife, Mrs. Kagiso Baatseba Imuede, I love you, and I am always grateful for the peace you effortlessly exude.
- My father and siblings, your love, moral support, and motivation are never-ending.
- Finally, my invaluable appreciation goes to the Financial Aid & Scholarships Office - WITS University for awarding me the Postgraduate Merit Award (PMA) bursary for two years. Your investment in me is a direct reflection of a successful study completion. THANK YOU!

Conference Presentations & Publications

Segments of this research have been presented and published as stated;

- J. Imuede, P. Ranchod and M. Raborife, Applying AI and Web Services in Mining Sexual Violence Tweets in South Africa. Paper and Poster accepted for presentation at the 3rd Black in AI Workshop (co-located with NeurIPS 2019) at the Vancouver Convention Center, Vancouver, Canada.

- J. Imuede, P. Ranchod and M. Raborife, Inferring Gender and Sentiment Analysis from Streaming Sexual Violence Tweets in South Africa (Abstract and Poster). Accepted and to appear in the proceedings for a virtual presentation at the 3rd MD4SG & ACM FCRC co-located conferences, June 2019, Phoenix, USA.

- J. Imuede and P. Ranchod (2019). Mining Sexual Violence Tweets in South Africa using Premade TensorFlow Estimator (Abstract and Poster). Accepted for presentation at the Deep Learning IndabaX Conference on AI & ML, Durban, South Africa.

- Jude Imuede (2018). Mining Tweets on Sexual Violence in South Africa. Abstract and Poster accepted for the 12th CHPC National Conference on Transforming the Future through HPC and Transforming HPC for the Future, Cape Town [South Africa] https://events.chpc.ac.za/event/33/contributions/729/contribution.pdf

- J. Imuede, M. Raborife and P. Ranchod (2018). Extracting Demographic Attributes from Twitter Using Sexual Violence related Tweets, In Proceedings of the 2nd International Multi-Disciplinary Conference (IMDC), Lusaka, Zambia.

- The Eighth Cross-Faculty Postgraduate Symposium (2017) at WITS University. Mining Tweets on Sexual Violence in the Inner-City of Johannesburg. Poster accepted in Showcasing Postgraduate Research at the University of the Witwatersrand, Johannesburg, South Africa.

Contents

Declaration iii

Abstract v

Dedication vi

Acknowledgements vii

Conference Presentations & Publications viii

Table of Contents ix

Contents . . . ix

List of Figures . . . xiii

List of Figures xiii

List of Tables . . . xvi

List of Tables xvi

1 Introduction 1

1.1 Introduction . . . 1

1.2 Mining Twitter Data . . . 2

1.3 Motivation and Rationale . . . 2

1.4 Background . . . 3

1.4.1 Problem Description . . . 4

1.4.2 Research Question . . . 4

1.4.3 Research Context . . . 4

1.4.3.1 Aim(s) . . . 4

1.4.3.2 Objective(s) . . . 4

1.5 Research Methodology . . . 5

1.6 Results . . . 6

1.7 Contribution of the Study . . . 6

1.8 Summary . . . 7

1.9 Dissertation Layout 7

2 Background and Literature Review 9
2.1 Introduction 9
2.2 Sexual Violence 10
2.2.1 Landscape of Sexual Violence in South Africa 11
2.3 Overview 12
2.3.1 Twitter 12
2.3.1.1 Twitter Geo Search for Mining Tweets in South Africa 13
2.3.1.2 Twitter API 14
2.3.1.3 Twitter's Streaming API 15
2.4 Sentiment Analysis 15
2.5 Feature Selection 17
2.5.1 Transfer Learning 17
2.6 Gender 18
2.6.1 Gender Prediction 18
2.7 Classification Measure 21
2.7.1 Confusion Matrix 22
2.8 Overview 24
2.8.1 TensorFlow API 24
2.8.2 TensorFlow Serving 26
2.9 Elasticsearch and Kibana: Storage and Analysis 27
2.9.1 Use Case 1: 27
2.9.1.1 Architectural Description 28
2.9.1.2 Data Extraction and Store 29
2.9.1.3 Data Search and Visualisation 29
2.9.2 Use Case 2: 30
2.9.3 Use Case 3: 31
2.10 Summary 34

3 Research Approach and Methods 35
3.1 Introduction 35
3.2 Assumptions 35
3.3 Core Frameworks 36
3.3.1 Gender Predictor 36
3.3.2 Sentiment Analyser 37
3.4 Research Methods & Implementation 38
3.4.1 Indexer 40
3.4.2 Code Design for Gender Predictor 40
3.4.3 Data Overview 42
3.4.4 Feature Engineering 43
3.4.4.1 Encoding the Target Variable (Gender) 43
3.4.4.2 Word-Embeddings for Feature Columns 43
3.4.4.3 Train and Test Split 44
3.4.4.4 Build Input Function 44
3.4.4.5 Instantiate the Model 45

3.4.4.6 Train the Model . . . 46
3.4.4.7 Evaluate the Model . . . 46
3.4.4.8 Model Prediction . . . 47
3.4.4.9 Saved Model . . . 47
3.4.4.10 Visualising Loss and Model Graph with TensorBoard 48
3.4.5 Elasticsearch . . . 49
3.4.6 Kibana . . . 49
3.5 Summary . . . 49

4 Evaluation and Results **50**
4.1 Introduction . . . 50
4.2 Data Set . . . 50
4.2.1 Rationality of Results . . . 52
4.2.1.1 Analysing the Model . . . 52
4.2.1.2 Optimizing the Model for Serving . . . 54
4.3 Model Evaluation on Test Data . . . 55
4.3.1 Confusion Matrix . . . 56
4.3.1.1 Misclassification . . . 58
4.4 Summary . . . 58

5 System Review and Discussion **59**
5.1 Introduction . . . 59
5.2 The System . . . 59
5.2.1 Discover . . . 60
5.2.2 Visualize . . . 62
5.2.3 Dashboard . . . 64
5.3 Discussion . . . 65
5.3.1 Textual Discussion . . . 65
5.3.2 Visual Discussion . . . 66
5.4 Rationale of Gender Prediction and Sentiment Analysis . . . 69
5.4.1 Inferring Gender . . . 70
5.4.2 Inferring Sentiment Analysis . . . 70
5.5 Summary . . . 70

6 Conclusion **71**
6.1 Conclusion . . . 71
6.2 Recommendation for Further Study . . . 72
6.2.1 Technology . . . 72
6.3 Contributions . . . 73

Appendix A Pre-training the Gender Classifier **75**

Appendix B Helper Functions **80**

Appendix C Streaming Sentiment Prediction **84**

Appendix D DETAILS - INTEGRATING NODE.JS WITH PYTHON THROUGH TF-SERVING SERVER **87**

Appendix E EVALUATING TRAIN-TEST-SPLIT-SETS ON THE GROUND TRUTH DATA **90**

Bibliography **99**

List of Figures

1.1 Research flow of thought 5

1.2 Dissertation layout 8

2.1 The confusion matrix for a classification algorithm [Hamel, 2009] 22

2.2 An example of a PR curve analysis using manually-annotated entities between Loc-Text vs. Baseline [Cejuela et al., 2018]. 24

2.3 A "schematic TensorFlow dataflow graph for a training pipeline, containing sub-graphs for reading input data, preprocessing, training, and checkpointing state" [Abadi et al., 2016]. 26

2.4 TensorFlow Serving Architecture 27

2.5 Architectural description of collected and analysed flow information [Lahmadi et al., 2015]. 28

2.6 A piechart showing patterns of Android environment on a Facebook application [Lahmadi et al., 2015]. 29

2.7 An embedded system showing levels of attack [Nagara et al., 2017]. 31

2.8 Dashboard from value change alarms data [Hamilton et al., 2018]. 32

2.9 Collection of information through Logstash[Hamilton et al., 2018]. 33

3.1 The framework for gender predictor 37
3.2 The framework for the sentiment analyser 38
3.3 The overall system pipeline 39
3.4 The client-server architecture "HIGH-LEVEL" 40
3.5 The architecture for gender prediction (not all nodes in the hidden layers are represented) 41
3.6 Code block to instantiate the model to choose between two classes 46
3.7 Function to train the model 46
3.8 Snippet for model evaluation 47
3.9 Snippet showing model prediction 47
3.10 Function to save TF model 48
3.11 Model architecture showing computational graph 48

4.1 Graphs on model1 showing accuracy and loss 52
4.2 Graphs on model1 showing accuracy and loss on the validation set 53
4.3 Graphs on model2 showing accuracy and loss 53
4.4 Graphs on model2 showing accuracy and loss on the validation set 53
4.5 Graphs on model3 showing accuracy and loss 54
4.6 Graphs on model3 showing accuracy and loss on the validation sets 54
4.7 Confusion matrix on the training data 57
4.8 Confusion matrix on the testing data 57

5.1 The Kibana's interface showing cumulative estimates of gender variables and sentiment scores 61
5.2 Real-time tweet sample from the Discover menu bar 62
5.3 Feature extractions from the Kibana menu 62
5.4 Graphical representation showing menus of visualizations 63

5.5 Male contribution on sexual violence tweets 63

5.6 Female contribution on sexual violence tweets 64

5.7 Dashboard 1: This reflected on the city with most reported cases in terms of gender and sentiment represented 64

5.8 Dashboard 2: This presents the handles in South Africa with most frequency of tweets indicated in bold 65

5.9 These are some of the cities showing origin of tweets 66

5.10 A Pie chart showing contributions from male and female 67

5.11 A Heat map showing sentiment count 68

List of Tables

2.1 The overview of the Twitter API 15

4.1 Train matrics after training using the DNNClassifier 55

4.2 Test matrics after testing using the DNNClassifier 55

4.3 This is an excerpt from our confusion matrix in Figure 2.1 56

5.1 Columns on feature extractions from Kibana 62

5.2 Sentiment count from both gender classes 68

5.3 A description of relative fields from tweet document 69

List of Abbreviations

AFINN	Finn Arup Nielsen
AI	Artificial Intelligence
API	Application Programming Interface
BackProp	Back Propagation
CSV	Comma Separated Value
DFFNN	Deep Feed Forward Neural Network
DL	Deep Learning
DoS	Denial of Service
DNN	Deep Neural Network
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GBV	Gender-based Violence
GUI	Graphical User Interface
GPU	Graphical Processing Units
HTTP	Hyper Text Transfer Protocol
IoT	Internet of Things
JS	Java Script

JSON	JavaScript Object Notation
KNN	k-Nearest Neighbours
LIWC	LIWC - Linguistic Inquiry and Word Count
ML	Machine Learning
NBM	Naive Bayes Multinomial
NN	Neural Network
OOV	Out of Vocabulary
PCA	Principal Component Analysis
PMI-IR	Pointwise Mutual Information and Information Retrieval
PR	Precision-Recall
REST	Representational State Transfer
SSL	Secure Shell Layer
SVM	Support Vector Machines
TF-Hub	TensorFlow Hub
TF-Serving	TensorFlow Serving
TL	Transfer Learning
TPU	Tensor Processing Units
TF	TensorFlow
TP	True Positive
TN	True Negative
URL	Universal Resource Locator
UI	User Interface
USA	United States of America
WHO	World Health Organisation

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

This chapter introduces the need for looking at the problem of Gender-based Violence (GBV) by mining Twitter data and provides a motivation through emphasising the need for automated estimates of gender representation and sentiment expression from GBV tweets in real-time. The chapter then discusses using disparity in gender and sentiment in reported cases as a measure to analyse patterns of occurrence based on geolocation. It is envisioned that this investigation packaged as an interactive data visualisation tool can be used by policy-making and law-enforcement agencies to implement suitable preventive measures and policies. The reason is that we attempt to address the rampant nature of sexual violence in South Africa, and the robustness of our adopted Deep Neural Network (DNN) technique emphasises that our approach is timely, viable and appropriate for mining data on this subject. In particular, we are interested in inferring hidden demographic attributes from Twitter by training an algorithm to predict if a tweet on sexual violence belongs to either a male or a female. With that in mind, we built a DNN using a pre-made estimator (DNNClassifier) from TensorFlow high-level API to perform gender classification and prediction. Moreover, we implemented a score-based sentiment analysis using the Finn Arup Nielsen (AFINN) word list to extract sentiment from a given tweet by way of polarity measure. From there, we went on to visualise our results on the Web using services from Elasticsearch and `Kibana` through the implementation of `Node.js`.

1.2 MINING TWITTER DATA

Mining Twitter data provides a significant source of insight generation when applying machine learning (ML) techniques from the field of deep learning [Wongsuphasawat et al., 2018]. Twitter data is now ubiquitous with research interests cutting across applicable domains in artificial intelligence (AI) and ML. The effect of mining Twitter is a data-driven transition in technology and a subfield in knowledge management which helps in information discovery given large amounts of unstructured data. As we have come to acknowledge from this research, the problem has never been the kind of data to be investigated but rather, selection of a preferable approach and model which can be explored to answer questions and generate insights from the data. By definition, Twitter is a web service which continuously attracts millions of internet users with content for mining and inferring socio-economic indicators and demographic attributes [Reips and Garaizar, 2011]. The platform offers a vast amount of unstructured data on public opinion for research purposes. According to Reips and Garaizar [2011], Twitter has continually shown prominence as a data source hub for researchers and data science enthusiasts.

Over the last few years, South Africa has experienced strong acceptance and rapid growth in the use of Twitter, albeit not at the rate at which it is seen in some other major developing countries [Berthon et al., 2012]. Possible reasons could be: increasing access to the Internet, wireless communications and several industralised cities hence, providing a basis for choosing the country for Mining Twitter Data.

1.3 MOTIVATION AND RATIONALE

Gender-based Violence (GBV) is a current reality which has long existed and has a global impact on society. Broadly speaking, sexual violence is one form of GBV, described as violence that arises as a consequence of normative role expectations and unequal power relationships that are forcefully imposed based on gender. GBV is a serious and prevalent issue in South Africa, affecting almost every aspect of life [Dunkle et al., 2004]. In particular, According to the South African Police Services and Statistics South Africa respectively, 80% of reported sexual offences were rape and 68.5% of the sexual offence victims were women in the 2016/17 year - *One of the highest figures in the world* [StatsSA, 2018]. The World Health Organisation (WHO) defined sexual violence as:

> "*any sexual act, attempt to obtain a sexual act, unwanted sexual comments or advances, or acts to traffic, or otherwise directed, against a person's sexuality using coercion, by any person regardless of their relationship to the victim, in any setting, including but not limited to home and work.*"

Overall, sexual violence is an issue with enduring negative impacts on victims and the society at large. Regardless of the perspective at which this problem is conceptualised, the main essence of sexual violence counteractive action is simply to prevent it from occurring in any case. The solutions, be that as it may, are similarly as complex as the issue. Ultimately, preventive measures should focus on drastically reducing the statistics on perpetrators and victims alike with the aim of reducing risk factors and advancing defensive mechanism.

With this issue in mind, we propose an interactive, real-time data visualisation system that can support information-based journalism, policy and law enforcement. The envisaged system could provide perspective on what sexual violence means to different gender classes, assist in identifying some of the causes of sexual violence through highlighting similarities between reported cases, and identify key personas in the fight against GBV.

1.4 BACKGROUND

Twitter, unlike Facebook and Google+, do not provide the gender of its users, making gender a hidden attribute that needs to be inferred intelligently. The name of a person tends to be the preferred indicator of a persons gender in most languages [Fink et al., 2012]. However, name data on Twitter may not be reliable making it necessary to infer gender using other available user information.

Mislove et al. [2011], considered geographical distribution of Twitter users and highlighted gender and ethnicity in the United States of America (USA) based on a set of over one billion tweets collected between 2006 and 2009. They determined gender by matching the first name in the name field of the user profile with a list of popular baby first names born in the USA from data of the Social Security Administration. Rao et al. [2010] developed models which successfully predicted the gender of Twitter users by implementing a support vector machine (SVM) algorithm on socio-linguistic features like, n-gram characteristics and emoticons. This study exposed that inferring gender from user content produced a boost in prediction accuracy.

Unlike other related literature focused on inferring gender from tweets, our approach uses word-embeddings for feature selection and then infers gender using a Deep Neural Network (DNN) model. Essentially, the DNN model learns gender based on the users linguistic style and choice of expression. Moreover, it evaluates sentiment using a score-based system that is based on a list of words known as the Finn Arup Nielsen AFINN 111 technique Nielsen [2011b].

The inferred gender representation and evaluated sentiment expressed by the user are then visualised in an interactive platform together with the geolocation.

1.4.1 PROBLEM DESCRIPTION

To our knowledge, the issue of sexual violence as a form of GBV in South Africa has long existed both physically and digitally, yet no effective solution within a data-driven context helps tackle this problem. Especially, with the size and continuous growth of Twitter data posing a constant computational challenges in designing and implementing effective data mining in the real-time processing context. Moreover, the development of an interactive web-based, AI-driven system that can be used to investigate the disparity in reported cases of sexual violence in real-time using location, inferred gender and sentiment expressed from unstructured data is a big challenge because this data is unknown in terms of pattern, value, and insight.

1.4.2 RESEARCH QUESTION

Considering what is possibly achievable through ML algorithms and the application of a DNNClassifier, the question is as stated:

How can we use disparity as a metric for evaluating reported instances of sexual violence in South Africa using inferred gender and sentiment scores?

1.4.3 RESEARCH CONTEXT

1.4.3.1 AIM(S)

The perpetration of GBV is carried out by either a male or a female. This research is carried out considering this premise. Based on multiple tweets posted on Twitter relating to *GBV*, the main aim of the research was to analyse such cases within the context of sexual violence to provide stakeholders such as police, prosecutors and related personnel with a real-time intuitive information system that is useful in the execution of their jobs, for example, policymaking and policing.

1.4.3.2 OBJECTIVE(S)

Our objective is to use the disparity of reported instances of sexual violence in terms of inferred gender and assigned sentiment scores to provide an interactive visual interface supported by real-time data streaming and AI and show variations based on geolocation. In particular:

1. To develop a DNNClassifier through TensorFlow Estimator to classify, predict and evaluate gender variables for measures of correctness. These measures are implemented through the

applications of TensorBoard and Confusion matrix to evaluate the model's internal structure and performance, respectively.

2. To design an interactive user interface (UI) through the services of `ES` and `Kibana` that attempts to collect, store, analyse and visualise relevant tweets on sexual violence.

3. To infer sentiment analysis from the AFINN model of English word list which have been rated for valence. This will help us determine our measure of disparity of reported cases (tweets) by either gender class.

1.5 RESEARCH METHODOLOGY

The research objectives are achieved by developing a Web-based application for data inferencing and analysis. The application is focused on ingesting, preprocessing, annotating, analysing and visualising tweets on sexual violence. The Web-based application through the `Indexer` uses ES to visualise results in `Kibana` as indicated in Figure 1.1.

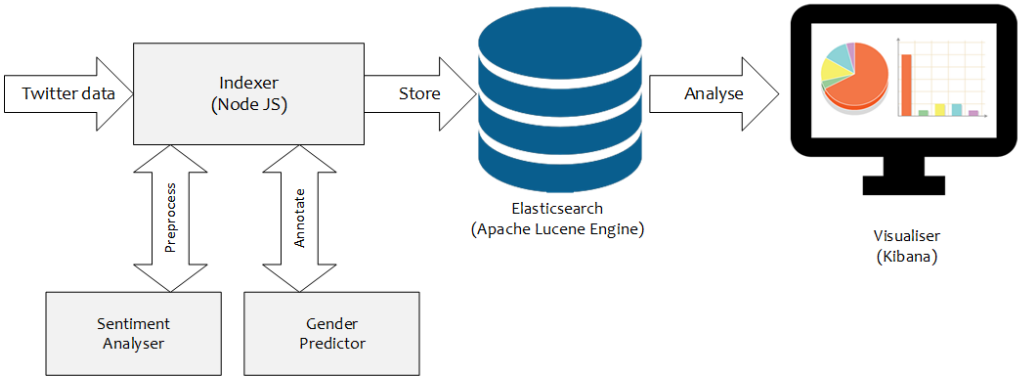


FIGURE 1.1: Research flow of thought

The methodology, which will be further expanded on in Chapter 3 through Figure 3.3 is achieved in these stated phases:

1. DATA COLLECTION: An application called the `Indexer` built on `Node.js` was used to connect Twitter streaming API to collect tweets found in the boundaries of sexual violence.

2. INFERRING GENDER: Inference was achieved using a gender predictor. This was made possible through a deep feed forward neural network (DFFNN) built using Google's TensorFlow Estimator. The training of the DFFNN was through the back propagation algorithm on labeled tweets from the gender classification datasets available on Kaggle. We used this data in training and verifying the model's performance.

3. INFERRING SENTIMENT: This phase of the research used a polarity-oriented approach in deducing the actual sentiment of a particular tweet text. We used a lexical resource; in particular, the AFINN word list which is a list of valence-classified English words scoring between negative 5 to positive 5.

4. DATA VISUALISATION: `Kibana` was used through `ES` to retrieve, analyse and visualise the annotated data.

1.6 RESULTS

The performance on gender prediction through TF Estimator has been evaluated experimentally. We connected the model's directory to TensorBoard to visualise the model's internals for analysis on accuracy and loss. The loss curve decreased over a pre-defined number of steps. This pattern confirmed that the model generalised well on out-of-sample data when tested. The accuracy (68%) on the test set returned a very good evaluation on the model from parameter and hyperparameter tuning. This is an expected condition for L1 & L2 regularisation. The sentiment analyzer, when tested with strings related to sexual violence, returned an efficient output but slightly declined when tested to negate an instance. Considering its computational performance, there is room for improvement. The disparity as captured from our system concludes that women are more negatively impacted by a measure of 72.74% in comparison to their male counterpart who only reported 65.57% based on sentiment. This makes sense in real-world and provides a good validation of our system as we would expect more negative comments from women than men Flood [2019], since women are disproportionately affected by sexual violence as noted in StatsSA [2018] where 250 out of every 100 000 women were victims of sexual offences as compared to 120 out of every 100 000 men in the 2016/17 year.

1.7 CONTRIBUTION OF THE STUDY

This research sets to describe our input in mining Twitter data for the following reasons:

1. To assist stakeholders such as law enforcement agencies to facilitate investigations regarding gender-based violence cases in South Africa by developing a real-time (Web-based) application for streaming sexual violence tweets.

2. Using the disparity of reported cased in gender and sentiment to find locations in which sexual violence is prevailing i.e. "hot areas".

3. To develop a model that motivates mining data on topics not related to sexual violence but of social relevance hence, improving or augmenting traditional methods with AI.

1.8 SUMMARY

Presented in this chapter is the aims and objectives, motivation and methodology of this research. The uniqueness of our use case from its complexities informed that we developed a model within the core estimator framework to specifically analyse tweets on GBV in South Africa. Analysing tweets on GBV in this instance meant that we utilised Google's TensorFlow high-level API (DNNClassifier, Estimator, TensorBoard) and the AFINN model before serving into production by using TensorFlow Serving and initialising Web services through the `Node.js` runtime environment. The main contribution of this research was to understand the disparity in gender representation and sentiment expresssion in South Africa using location through the development of a system that will help in visualising real-time sexual violence tweets and the implementation of AI and Web services using frameworks such as TF Estimator and AFINN model built on `Node.js` runtime. The next chapter expands on the topic through backgrounds and reviews of previous literature. Nonetheless, the layout of this dissertation is presented in the next Section of 1.9.

1.9 DISSERTATION LAYOUT

This dissertation is made up of six chapters including Chapter chapter 1 that provides the motivation for the research, the problem description, and the overall objectives of this study. The layout is schematically summarised in a flowchart in Figure 1.2. Chapter 2 is the background and literature review, which includes the previous knowledge related to this study amongst other details. Chapter 3 further extends on the methodology by providing the rationale for the research approach. The subsequent chapters (Chapter 4 through 5) evaluates the model's performance and present results in form of graphs. Again, it focused on an overview of the system built on `Node.js`. It presents the actual output of the research through the inference of hidden demographic attributes such as gender and sentiments from GBV related tweets. The dissertation concludes in Chapter 6 with summary of the findings and recommendations. Appendices A through E gave detailed experimental results and important inclusions.

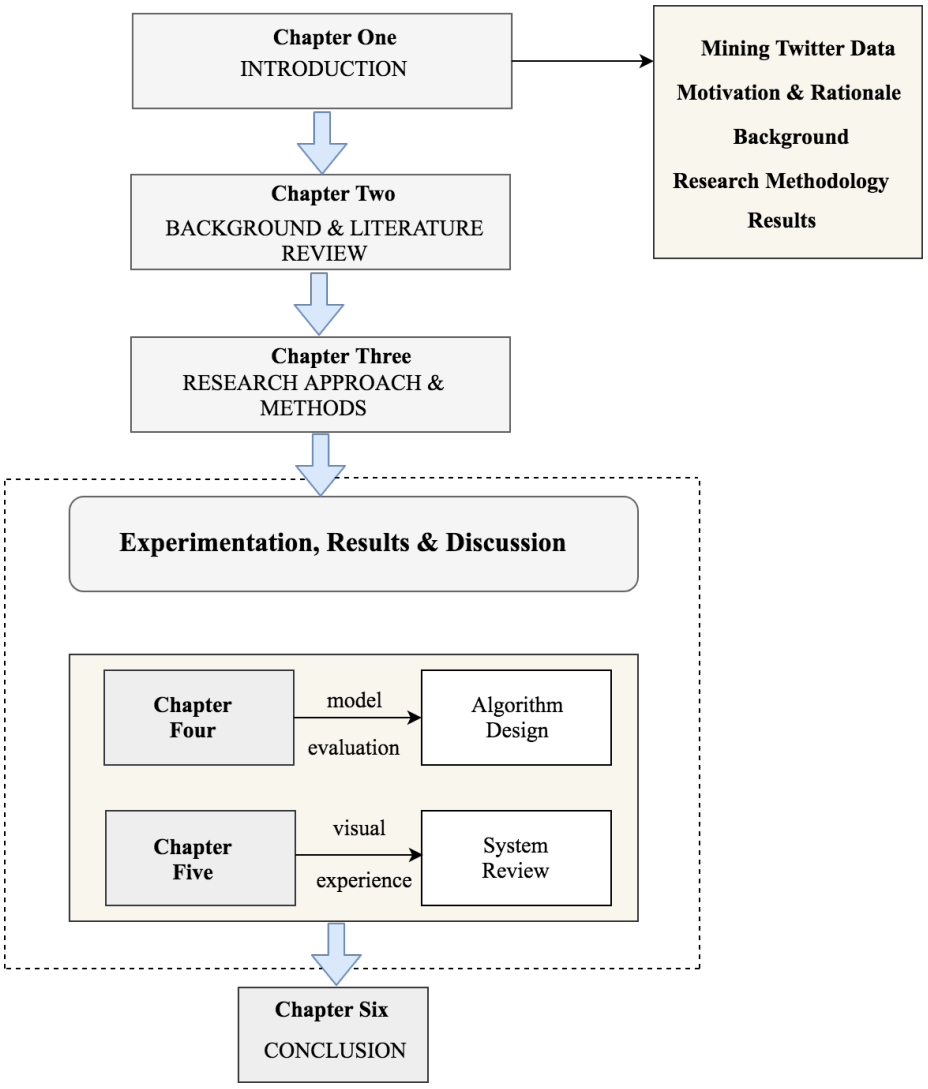


FIGURE 1.2: Dissertation layout

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

2.1 INTRODUCTION

Social media can be used to identify widely discussed topics. Twitter, for example, can provide an overview of currently trending topics and issues within and beyond the online space [Benhardus and Kalita, 2013]. According to Shah et al. [2018], the online processing of real-time data is rapidly becoming a key practice for analysing social media for socio-economical trends, politics, health, and social interest. Offline analytical procedures are traditionally only suggested when collecting data is a once-off process. Current trends in research demand sophisticated tools to tackle problems in real-time analysis of data, especially when it is related to a non-stop streaming of data collection where an efficient databases engine is required for such a purpose. Through this research, we try to develop a Web-based application to efficiently capture and address real-time challenges of sexual violence cases. We scientifically address a social science inspired problem powered by the community and fueled by data to contribute our AI-driven, Web-based application as a channel for further analysis.

The perpetration of gender-based violence (GBV) is carried out by either a male or a female. This research is carried out considering this premise. Based on multiple tweets posted on Twitter relating to GBV, the main aim of the research was to analyse such cases within the context of sexual violence so as to provide stakeholders such as police, prosecutors and related personnel with information that is useful in the execution of their jobs, for example, policy making.

GBV is disguised in different forms. Categories exist to classify these forms of relationship and the power exercised. The categories according to Edwards et al. [2014] are as follows:

1. *Rape and Incest.*
2. *Sexual harassment at work or in school.*
3. *Sexual violence against women in detention or in prison.*
4. *Violence against displaced women.*
5. *Trafficking in women; and*
6. *Domestic violence.*

It has been noted that human rights violations are committed against both men and women. The impact of these violations, however, varies according to the victim's gender. Studies such as Basow et al. [2007] have shown that the aggressive acts carried out on women differ, exhibiting certain traits, and therefore warrant being classified. This asserts that violence in this form can be linked directly to the unfortunate circumstance of power being distributed unequally. In addition to this, the devaluation of women and the subordination to men is due to the asymmetric relationship between men and women in society. The notion that women are the weaker sex is what separates GBV from other violent forms of aggression or coercion.

2.2 SEXUAL VIOLENCE

Sexual violence has become an everyday norm and therefore it has become harder for women to resist sexual violence [Kelly, 2013]. In most instances, such attempts end up with dire consequences for the women involved.

Power relations between men and women in South Africa are often characterized through sexual violence and assault [Organization et al., 2012]. Oddly enough, sexual health promoters are still figuring ways to address such a prominent issue happening every year with an estimated 1.3 million rapes occurring [Gilbert et al., 2014]. Power struggles in sexual relationships determine whether or not a woman protects herself against unwelcome sexual advances, pregnancy and STDs. On the contrary, the men involved determine the timing of sexual intercourse, the use (or not) of condoms and consequently whether a woman will conceive or not.

Violence against women far exceeds other forms of gender-based violence (GBV) in all provinces in South Africa [Jewkes et al., 2002]. Violence against women has been proclaimed a significant

concern for public health, social policy and human rights following "the 1993 World Conference on Human Rights and the Declaration on the Elimination of Violence against Women". It has been difficult to document the magnitude of violence against women as well as to produce reliable comparative data which can be used as a benchmark to monitor progress in this issue.

Devries et al. [2013] proposed two basic methods around synthesizing evidence to measure and analyse the prevalence of violence on an intimate partner as a form of GBV, which may be exhibited either emotionally, physically, or sexually. First, they systematically approached the review of global prevalence data from national or sub-national representative studies. They consulted 26 databases from medical and social sciences, analysed the world health organisation (WHO) global review on woman's health and reported domestic violence from 10 countries. Further, they conducted an international survey on violence against women involving 8 countries; to ascertain variables of choice, they analysed data from 16 countries focusing mainly on gender, culture and alcohol intake. To further estimate prevalence, they carried out a demographic and health survey in 20 countries.

The second method used by Devries et al. [2013] to estimate the rampancy of intimate partner violence in the lifetime of women was classical meta-regression. Twenty-one global regions were designed and adapted to different narratives in 2010, reflecting age-standardised estimates for country-specific and demographics on gender.

Using data from 141 studies in 81 countries, an estimate of the results was provided by Devries et al. [2013]. These studies revealed data on physical or sexual partner violence that occur at different levels of severity and for a different age group. Estimates from their model indicated that during their lifetime, women aged 15 and older experienced this kind of violence. There are significant variations in the rampancy of kind of violence across the region. We understand the cause and effects influencing the perpetrated instances highlighted in various strata of the study from the analysis of data collected. However, we hope that our approach through the creation of the proposed system will facilitate a more comprehensive solution in this regard.

2.2.1 LANDSCAPE OF SEXUAL VIOLENCE IN SOUTH AFRICA

An unfortunate reality is that women in South Africa are the prime target of gender-based violence (GBV) and they rarely have a protective option of negotiating safer sex [Wojcicki, 2002]. Also, due to the patriachal society that exists, man hold the power and women are largely excluded [Pitcher and Bowley, 2002].

It is essential, however, to mention that South Africa has a high prevalence of key rape risk factors, with many men currently experiencing trauma in childhood and a strong gang ideology. These problems further validate the abuse faced by women in many societies. Sexual violence is not

peculiar to South Africa however, a consequence of the post apartheid era. As a result, Soweto, along with some informal townships in the south of Johannesburg became known as the rape capital of the world [Mabasa, 2009]. However, race should not be considered as a valid indicator, as rape cuts across all racial groups and enclaves [Holzman, 1996]. The rampant rape in this area is a reflection of the distructive culture that was breaded by apartheid in these communities.

Countless acts of sexual violence remains unreported and unspoken of, not only to law enforcement agencies or institutions, but families and associates of the victims. It is estimated by the Medical Research Council (MRC) that the amount of sexual violence may be up to nine times the number of reported cases [Jewkes and Morrell, 2010]. We are invariably limited to the statistics of reported cases which certainly would not come in exact estimation of sexual violence prevalence in South Africa. The South African Police Service (SAPS) stated that in the year 2017/2018, 7,071,500 sexual violence cases were accounted for as complaints brought forward by victims or families of victims. Arguably, it is expected to implore critical thinking towards the figure of sexual violence in South Africa in that if the genuine figure is closer to 643,500 per year, it would imply that someone is sexually violated or indecently assaulted every minute in South Africa. In any case, with such a large number of sexual violations going unrecorded, and with recorded cases of violations being comprehensively categorised as 'sexual offenses', it is nearly impossible to ascertain the genuine nature and degree of sexual violence in South Africa.

The landscape of sexual violence in South Africa is highly unacceptable. This is expressive in its extent of predominance which is yet again, highly unknown but to a large extent can be controlled if not totally eradicated.

We emphasied that this research is not about clarifying the genuine nature of sexual violence as against other forms GBV, but the deepened reality of this kind of violence in its wide spread nature through tweets on Twitter. We attempt to approach this subject on sexual violence as a form of GBV from a scientific and analytic view through the use of AI and Web services in contributing our insight on mining Twitter data for insight generation.

2.3 OVERVIEW

2.3.1 TWITTER

The past few years have shown Twitter as a wonder in social media discovery with success in explosive growth, attraction of famous personalities and an increasing height in media and blog coverage [Graber and Dunaway, 2017]. Twitter's fluidity and dynamism cuts across universal boundaries with its millions of users almost tirelessly tweeting. Arguably, it is among some of the well known and frequently used social media platforms with dual functionalities as a micro-blogging servicing

site and a rich social media data-generating platform [Greenwood et al., 2016]. With its daily generation of terabytes of data as its users exponentially increase [Khan et al., 2018], billions of tweets are made available through its API offerings [Abbasi et al., 2014]. According to Longley et al. [2015], Twitter users post 40 million tweets on a daily basis. As of January 2013, it is ranked as the most popular site in the world with 500 million registered users[1].

Twitter is a widely accepted microblogging service which encourages users to initiate status messages (called tweets). Often times, these tweets reveal impressions about issues on different topics [Ren et al., 2016]. Barberá and Rivero [2015] added that, Twitter is praised for its immediate information access in light of the promptness of immediate update in the 'twitterverse' of the user's profile, follower's and followings' timelines'. Other than the macro-level potentials for organisational and societal strides which are dependably lucrative Wagner III and Hollenbeck [2014] state that with a user base of that size, it is the fundamental system dynamics that made up the gravity for such a user base to emerge at the level it currently is. For this reason, Twitter is extremely popular and offers rich and fascinating views into the users' lives.

We consider a few potential mining possibilities of data from Twitter with users within the parameters of the stated categories. These are:

What are people currently talking about as of now?

Mining entities from a pre-defined user's tweets.

Sentiment analysis as related to peoples' opinion on a focused topic.

The Twitter following model is straightforward and yet it exploits a key part of what makes us human. We chose to mine Twitter data because of its richness in fluidity and dynamism with regards to its sophisticated mechanism that its users interact with below or within the 140 to 280 character limit as the case may be.

2.3.1.1 TWITTER GEO SEARCH FOR MINING TWEETS IN SOUTH AFRICA

The Twitter network is a global platform. However, sometimes, you may prefer to focus on users in predefined locations to achieve the goal of pulling tweets in a specific location with respect to keywords and search terms. Geolocated Web-based social media data presents as an affluent source of information about the place and provincial human conduct.

Twitter gives a ceaselessly-updated stream of data which has demonstrated to be valuable for predicting group concerns and modeling populations behaviour [Habermas, 2015]. Relating data with

[1] https:/www.alexa.com

the specific geolocation from which it originates makes an effective tool to model geographic phenomena, such as monitoring influenza, predicting elections, or observing and documenting linguistic differences among groups. Be that as it may, just a small size of data found on social media is accompanied by the location; for instance, a percentage less than one of Twitter posts provide geolocation [Jurgens et al., 2015].

In this manner, the latest work has concentrated on geo-inference for predicting the location of posts from persons [Jia et al., 2014]. One particular direction of geo-inference using social network has created approaches that claim to precisely identify the majority of posts within several kilometers of their actual locations [Miura et al., 2017]. In spite of the critical enthusiasm for geo-inference and reported achievement of strategies, looking at current evaluation practices uncovers geo-inference difference between evaluation settings [Backstrom et al., 2010].

Additionally, the conditions in which techniques have been tested frequently do not reflect this present reality conditions in which these strategies are required to operate. In specifics, the inclusion of a geo-location search into this research is so that we are able to ascertain the locations to mine Twitter data and its attributes. We consider using Twitter streaming API in pulling tweets around South Africa as a geographical landscape.

2.3.1.2 TWITTER API

According to Go et al. [2009], Twitter's Application Programming Interface (API) functions through written computer programs to allow access of data through querying. It is said to be the instruction to interact with some kind of technological parameters mainly for developers. Given the current situation in its open API mechanism, Twitter has a massive data volume and a liberal policy that allows external developers to build systems that rely on Twitter's data. The take-away from an open API offering is primarily to foster external innovation that further enhances base technology as a service improvement through access to its data. Giving access to external developers makes it possible to improve products, platforms and interfaces that have already been created without the need to uncover raw information. Instead of building systems internally, it is understandable for technology companies to make other imaginative advances. Twitter took advantage of this model as confirmed by its 2012 acquisitions of 10 different technology companies locked around its open API[2]. Besides the abridged stated possibilities which we have considered for this research, there are three different ways to access data from Twitter which are summarised below in Table 2.1:

The criteria may be keywords, username, tokens, geographical locations, etc.

Twitter has a quota limit on the amount of tweets that can be returned in one request for the Stream and Search APIs.

[2]https:/www.investopedia.com

Twitter Limits the number of requests that can be made within a specific timeframe.

TABLE 2.1: The overview of the Twitter API

Twitter API	Description	Cost
Search	Allows searching for tweets that match a particular criteria	Free
Streaming	Allows users to get tweets in near real-time based on the criteria	Free
Firehose	Pushes data to end users in near real-time based on the criteria	paid

2.3.1.3 TWITTER'S STREAMING API

For this research, we made use of the streaming API for the extraction of sexual violence tweets through the ingestion of tweets by an `Indexer` built of `Node.js`.

The streaming API of Twitter is a push of data as tweets are created and posted in near real-time [Kumar et al., 2014]. This streaming API provides an estimate of all tweets corresponding to some predefined parameters like *keywords*, *username*, *geographical locations*, *tokens* etc. set up by the API user. Tweets that corresponds to those predefined parameters are pushed almost immediately to the user as requested [Morstatter et al., 2013]. The algorithm set is facilitated by Twitter through a stated parameter that a user receives tweets, almost immediately, that matches the keyword relating to a search-term like *rape*. As it happens in real-time, the result from the request will be delivered to the user, into a predefined database (in our case, `ES`). Twitter works with the streaming API using the push mechanism against the search API that works using the pull mechanism created from the end user's involvement to a certain topic. A major disadvantage of the streaming API is based on the fact that Twitter only accounts for a limited amount of tweets that are occurring in real-time. This drawback, which affects the percentage of expected tweets made available by Twitter in using the streaming API, is a result of the traffic at the instance when the request and the conditions were made. Twitter's restriction on streaming API is that users are allowed 1% of tweets on a real-time basis to over 40% of tweets. Currently, this API offers some disadvantages as it does not support a 100% push request made by a user. To circumvent this issue, Twitter developed the Twitter firehose which falls outside the scope of this research.

2.4 SENTIMENT ANALYSIS

The intention of the sentiment analysis is to determine the attitude of the speaker or writer towards a certain subject matter or the overall contextual clarity of a document. Different methods have been used in solving problems in sentiment analysis. For example, machine learning algorithms through

deep neural networks (DNNs) [dos Santos and Gatti, 2014] and the Bayesian classification [Fersini et al., 2014], have been explored from pre-trained datasets through a supervised approach.

We approach this problem with a pre-made list of assigned words that have a valence i.e, (positive or negative score) to specific words. This list is known as the AFINN 111 dataset and the newest version includes 2477 words and phrases labeled between 2009 and 2011 by Fin Arab Nielsen, all relating to GBV.

Previous research on Twitter for sentiment analysis has explored various methods, but there is still room for more research output in this field. Go et al. [2009] experimented on Twitter for sentiment analysis and identified the polarity of the tweet using emoticons as noisy labels through collecting 1.6 million tweets as training data. In their Naive Bayes classification, they reported an accuracy of 81.34%. Turney [2002] worked on the reviews of products, using two out of the eight parts of speech i.e, adjectives and adverbs for example, to classify opinions on assessments. Using 410 product reviews from different domains all collected from Epinion[3], the semantic orientation of the sentiment phrase was estimated using the PMI-IR algorithm. He was able to attain an average precision of 74%. Several ML algorithms have been tested by Pang et al. [2002] on Movie Reviews. In the Naive Bayes classification unigram, they achieved 81% accuracy.

In their paper on Mining Sentiments from Tweets, Bakliwal et al. [2012] introduced a method for finding opinions in a clustered group of tweets. With suggestive words being labeled as noisy, a list containing such words was used as one dataset while the other was built using emoticons. A method used for scoring polarity was proposed. Emphasis was given to types and levels of preprocessing which are necessary for good performance. The Naive Bayes method was utilised for the determination of the token polarity in tweets, this method performed well on both datasets, moreover, their process of mining sentiments from tweets was thorough. They used both the Stanford and Mejaj's dataset.

Pak and Paroubek [2010] in their study "Twitter as a Corpus for Sentiment Analysis and Opinion Mining" performed sentiment analysis through Twitter data. They worked on a corpus of data for analysing sentiment and opinion which resulted in the performance of linguistic analysis on the collected corpus. They were able to develop a sentiment classifier that was built to determine positive, negative and neutral sentiments. They used their proposed technique on English language and recommended it could be used with any other language. Their technique was observed to have performed more efficiently than previously proposed methods.

In this research, we used a word list model which is bench-marked on Nielsen [2011a].This model is polarity-oriented and involves a pre-assigned list of words. Each word gets a positive or negative valence score, 5 being very positive and -5 very negative regardless of language. As indexed in

[3]https://en.wikipedia.org/wiki/Epinions

Appendix B and C, we showed the building blocks of sentiment analysis before it was feed into the `Node.js` run time environment.

2.5 FEATURE SELECTION

This project implements feature selection through extracting tweets using search tokens and keywords such as “rape”, “sexual abuse”, “sexual assault”, “sexual harassment” and “sexual violence”.

In a paper titled “Feature selection for classification”, Dash and Liu [1997] asserts that computational performance has greatly improved with lots of work done on feature selection for text analysis. As the development of large databases and requirements for good machine learning techniques continues to grow, a rise in new problems and feature selection approaches is observed. A study from OâKeefe and Koprinska [2009] on “Feature Selection and Weighting Methods in Sentiment Analysis” confirmed that researchers are still having challenges in employing basic feature selection techniques in a bid to improve the performance of their work through the use of complicated approaches. In their study, they reiterated that in improving sentiment analysis, only two papers have really expatiated on feature selection. Pang and Lee [2004] worked on the first paper based on subjective and objective text - trained support vector machine (SVM) to remove objective sentences found in the corpus. They discovered in their initial results that the accuracy of the classification of sentiment document actually declined. They conducted a method around hidden feature engineering on the basis of assumptions which involve sentences adjacent to removed sentences. This improved their accuracy level over their baseline. The second improvement of sentiment analysis was an attempt to scale the level of accuracy as performed by Abbasi et al. [2008] using a sophisticated feature selection technique. From their finding, the discovery of improvement in accuracy was demonstrated and attributed to either information gain (IG) or genetic algorithms (GA). They developed a hybrid algorithm from the combination of IG and GA called the Entropy Weighted Genetic Algorithm (EWGA), which resulted in achieving the highest level of accuracy of sentiment analysis 91.7% to date. The disadvantage of this algorithm is the initial selection of features which is computationally very expensive for both IG and GA.

2.5.1 TRANSFER LEARNING

The Transfer learning (TL) method takes the weights and variables of a pre-existing model that have been trained on a large volume of data and leverages on it for a prediction task.

In many fields of knowledge engineering, including classification, regression and clustering, data mining and ML, technologies have already accomplished significant achievements [Wu et al., 2008; Yang and Wu, 2006]. Many ML techniques, however, only function well under a prevalent premise:

training and testing data are derived from similar feature space and distributions. Using freshly gathered training data, numerous statistical models require to be re-generated from the begining when the distribution changes. In many real-world applications, is expensive or impossible to collect the training data and rebuild the models. Thus, it is preferred to reduce or eliminate these laborious efforts of collecting the training data. In instances such as this, it would be beneficial to transfer knowledge between task domains.

The applicable concept of TL is conventional with models on computer vision, but has a limited application to text classification, essentially, in the case of tweet document. We used TensorFlow Hub[4], a machine learning library to improve TensorFlow models with TL on gender prediction.

Based on previous literature, many examples can be discovered in knowledge engineering where TL can be of real benefit. A study by Pan and Yang [2009] conducted in this regard was an instance of classification of Web documents, where their aim was to categorize a specified Web document into various well defined divisions. By way of an analogy, Patricia and Caputo [2014] discussed a situation that concerns an area of classification of Web documents. It was noticed that the marked examples were university web pages associated with grouped information obtained through manual labeling. They restated that, there may be a inadequacy of labeled training data for a classification task on a newly generated website where the data features or data distributions may be different. As a result of this, there was a limitation; the web page classifiers learned on the university website could not be applied to the new website. As such, it was suggested that it would be helpful if the classification knowledge could be transferred into the new domain.

The advantage of TL is mainly that you do not have to provide as much data about your own learning algorithm as you would if you were to build from scratch. The pre-existing models, which is a cloud-based repository of large volumes of already trained models in TensorFlow Hub (TF-Hub) provides checkpoints for various types of models - audio, images, text, and more. However, in this research, our concern is centred on TL for classification task that is related to data mining problem. To predict the gender of Twitter users in South Africa from author's tweets and profile description using TF-Hub module for text with the intention to improve generalisation and speed up the learning rate, a predictive model was built.

2.6 GENDER

2.6.1 GENDER PREDICTION

This research is driven by its positive impact in analysing sexual violence related cases from tweets in South Africa through the exploration of deep learning (DL) techniques and the application of

[4]https://www.tensorflow.org/hub/

web services for social good. In relation to Twitter data, we have seen from various literature reviews the application of DL techniques by Badjatiya et al. [2017], Nguyen et al. [2016], Heck and Huang [2014], Severyn and Moschitti [2015] and Yuan et al. [2016]. However, none of these stated reviews reflect issues similar to the South Africa context in terms of inferring gender from sexual violence tweets and the method used. The concept could be said to be the same in terms of its "deep learning" approach, but that is as far as the similarity goes. As a result, we look at related work on gender prediction.

Fink et al. [2012] found that demographics like gender and ethnicity adversely affected the way people react and respond to the same subjects of discussion at that moment. Therefore, to infer gender from only the content of tweets as features, a region-specific study was conducted to train an algorithm that classifies tweets using supervised machine learning. When a combination of all these features were used rather than just the unigrams, they yielded significantly better results with an F-score of 80%. Furthermore, the addition of hashtag and Linguistic Inquiry and Word Count (LIWC) showed no significant improvement to the performance by directly using unigrams. When the features were evaluated individually, the hashtag feature performed badly giving low F-scores while the LIWC showed a slightly better performance. However, both their performances were still reasonably lower than those of unigrams. To access the classification results of the unigrams, Fink et al. [2012] first investigated which features were better suited to distinguish between genders by making use of each feature's result or each feature from Monte Carlo results and calculating the mean of the weighted sum of all support vectors of each feature. Comparisons between the results revealed that the positive weights were good discriminators of gender as they were used to classify females while the negative weights were used for males. The unigram features that were used to analyse the classes identified significant emotive words and symbols that were used by the different genders. For example, females used words like "miss", "aww", sad and happy emoticons more while the males used more profanity and soccer references such as "game", "arsenal", "united" and "chelsea".

Through data collected from an online game, Nguyen et al. [2014] combined insights extracted from sociolinguistics and social indicators to highlight the importance of approaching gender from a social indicator point of view as opposed to a not a biological view point. They used their game as an inference benchmark with thousand of players predicting the gender of a certain Twitter account holder based on the linguistic pattern of tweets alone. The results reflected 10% inadequacy on Twitter users who do not use language associated with the crowd's biological identity. It was also seen that users who have been active for a long time on the network were often perceived to be younger. The overarching conclusion bothers on the authors' findings, which highlights the limitations of current approaches in predicting gender from text. They further indicated that their findings applied to social variables like ethnicity.

AlSukhni and Alequr [2016] worked on the gender of tweet authors through the application of various Classification of Arabic mining techniques. They used tools like Naive Bayes, support vector machine(SVM), Naive Bayes Multinomial (NBM), J48 decision tree, and K-Nearest Neighbours (KNN). They explored these machine learning classifiers to detect which gender communicated in an Arabic language through tweets. Their training sets contained 4017 tweets and validated the impact of pre-processing on classifier accuracy. This resulted in a negative output. Author names and word features had a significant positive effect in determining the accuracy level of various classifiers. An accuracy of 98% was achieved by these classifiers: J48, NBM, and SVM. The overall result of all classifiers in recall and precision has been remarkably improved with the measurement metrics. The combination of words and the average length of tweet words recorded a positive effect. KNN and Naive Bayes had a negative impact on precision. This demonstrated the impact of machine learning as a sufficient technique for predicting gender from the tweets made by Arabic authors. Their percentage of cross-validation and splitting resulted in the same findings when the experiment was prepared. As a conclusion, the classifiers of NBM, J48, and SVM performed better in classifying an instance of a female tweet rather than a male tweet.

Miller et al. [2012] used perceptron and Naive Bayes algorithms to identify the gender of Twitter users from 1 to 5 grams in tweets. Their work used streaming algorithms and made substantial use of gender prediction to manage speed and measure of tweet traffic. Features such as R-gram were implemented to represent streaming tweets because informal text (e.g. tweets) was not easily assessed using traditional dictionary methods. Multiple selection methods were used to select informative n-gram features as a large number of 1 to 5 grams requires only one subset to be used in the classification of gender. The Naive Bayes and Perceptron algorithms were 99% accurate. Six selection algorithms were implemented in extracting informative features and to improve the classification and run time of their gender prediction approach. The approach initiated through the stream mining perceptron and Naive Bayes performed relatively well to evaluate how effective these selected gender identification features on Twitter were. The perceptron functioned reasonably well with a very high precision of 97% and a balanced accuracy of 94% (this was outperformed by Naive Bayes with a score between 90% and 100% accuracy for all metrics).

Culotta et al. [2015] on "Predicting the Demographic of Twitter Users from Website Traffic Data" set out to predict users' demographics on the basis of who they are following. Audience measurement data from 1,500 websites was used to create a distantly labeled dataset. A regression model was then fit to the data which was then used in the prediction of six demographic variables (gender, age, ethnicity, education, income, and child status) of a set of bases for users on a following basis. Their approach was quite practical as they fit a model of regression linking sets of users to their demographic profile through the connection of data from web traffic with followers from Twitter. They arrived at a regresssion model which was based on the follower of Twitter over six

demographic variables predicted the average hold-out correlation of 0.77 between the web traffic demographics of a website. To address the question of whether a regression model could be extended to classify individual users, they found that the regression model competes with a fully supervised approach using a hand-labeled validation set of gender and ethnicity annotated users. They found that randomly selected identities of only 10 accounts followed per user were sufficient to achieve 90% of the accuracy obtained using 200 accounts followed. They performed cross-validation, fix-folding, and generated the hold-out correlation coefficient between the demographic variables predicted and the actual variables. By comparing multi-task elastic net with single-task elastic net and a ridge regression variant (with regularisation parameter optimisation), they derived a significant improvement from the ridge to the elastic net and a more modest improvement when using multi-task formulation. The regression results suggested that the adjacent account vector was highly predictive of the demographics of its followers. The relationship between user interest and perceived psycho-demographic attributes has been analysed using Twitter data from over 4000 Twitter user profiles. Models were trained to predict the personal traits of different users.

Volkova et al. [2016] took advantage of the fact that users are integrated into Twitter's social network as opposed to the existing work, which is based on predictions of users' textual tweets. They looked at their user's followed accounts to ascertain users to predict perceived personality traits. Percieved users demographic attributes were discussed. This includes: gender, age, education background, political status. They used crowdsourcing to annotate the user profile of their personal characteristics. They compared the method used in achieving the accuracy from personal triat prediction with approaches from state-of-the-art that rely majorly on tweets from users. The results of this work showed user interests related to the embedding of the social network of the user, and this can influence insightful hints on a user. They also found that features such as a user's interest area are a predictive pointers as some demographic characteristics from the user'e generated text. They found that observing who follows the user and what content they consume is sometimes enough to accurately infer many of the features of the user. For example, in those cases where a user's interest is ascertained from his browsing history, but does not contain any text produced by the user. We understood that a user's psycho-demographic profile can be predicted given these interests. These interest-based models provides an excellent alternative to the existing methods based on user communication.

2.7 CLASSIFICATION MEASURE

The measure of gender in a classification task is significant to arrive at what actual instance of a sexual violence tweet was made by either a male or female. We used the classification concept as a popular machine learning application in detecting the level of correctness of an algorithm. This section explains the measure of gender classification through the concept of a confusion matrix as a

measure of performance evaluation. This is basically to check the case of *true* or *false* in a complex task adddressing gender classification.

2.7.1 CONFUSION MATRIX

The application of confusion matrix to the field of machine learning for performance evaluation has been applicable to not only the domain of binary classification. Confusion matrix, also known as an error matrix is a table layout that shows the visual performance of an algorithm. A confusion matrix reveals information about predicted and actual instances as contained in a classification scenario Lewis and Brown [2001]. Each row of the matrix is a representation of the predicted class, and the columns reflects an instance of the actual class. Essentially, confusion matrix attempts to verify the misclassification rates between two classes. In this research, specifically in Chapter 4, we used a confusion matrix to further determine the measure of accuracy performed by the model. Considering a confusion matrix for a classification scenario, four metrics are used: *True positive*, *True Negative*, *False Negative* and *False Positive*.

		Actual cases	
		True	**False**
Predicted cases	**True**	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

FIGURE 2.1: The confusion matrix for a classification algorithm [Hamel, 2009]

Figure 2.1 showed the confusion matrix of a classifier from a gender classification model. In Figure 2.1, the metrics that appear along the diagonal of the Table are the correct classification instances stating variables from actual cases. The metrics are *True Positive* and *True Negative*. The metrics with incorrect classification is *False Positive (FP)* and *False Negative (FN)* which are also called Type I error and Type II error respectively. In achieving a model with 100% accuracy, *FP & FN* must be equal to zero. We now consider each metric in the context of gender classification algorithm.

- *True Positive (TP)*: The DNNClassifier correctly classified tweets as being tweeted by either a male or female.

- *True Negative (TN)*: The tweet was classified incorrectly as originating from female even though it was tweeted by a male or vice versa.
- *False Negative (FN)*: The tweet was classified incorrectly.
- *False Positive (FP)*: The tweet was classified as being from a male and was actually not an instance of a male tweet. But it was from a female.

In view of the stated metrics, a list of performance evaluation metrics can be estimated for a classification machine. As defined by Powers [2011], we present: *Precision*, *Recall* and *Accuracy* as the performance metrics.

- *Precision* defines the measure of correctness in the classification model of a positive instance (True Positive).

$$\text{Precision}\big(P\big) = \frac{TP}{TP + FP} \tag{2.1}$$

- *Recall* defines the measure of actual positive instances as identified in the classification model.

$$\text{Recall}\big(R\big) = \frac{TP}{TP + FN} \tag{2.2}$$

- *Accuracy* measures the aggregated instances of a correctly classified case.

$$\text{Accuracy}\big(ACC\big) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

We can see from the metrics that they evaluate a model's performance and are significant in evaluating a gender classification model in the case of this research.

As explained, a confusion matrix is a summary of prediction on a classification model. The number of correct and incorrect predictions are summarised with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which a given classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made by way of misclassification. Equally, it is used to generate a Precision-Recall (PR) curve [Ozenne et al., 2015]. *Recall* is plotted on the x-axis and gives an idea about when a classifier is actually positive and how often does it predict a positive. The *Precision* plotted on the y-axis of the graph on the other hand tells us about when it predicts positive and how often it is correct. The PR curve is also a metric for evaluating the performance of a binary classification that enables the visualisation of a classification model given a *thresholds* as a needed assessment for a classifier's performance. Figure 2.2 represents a PR curve used to compare the performance of a binary case. From the presented graph in Figure 2.2, the represented threshold is on the *x-axis* and *y-axis* of the graph.

When a classifier uses a threshold, the PR Curve shows the trade-off between false negatives and false positives as one move along the PR curve.

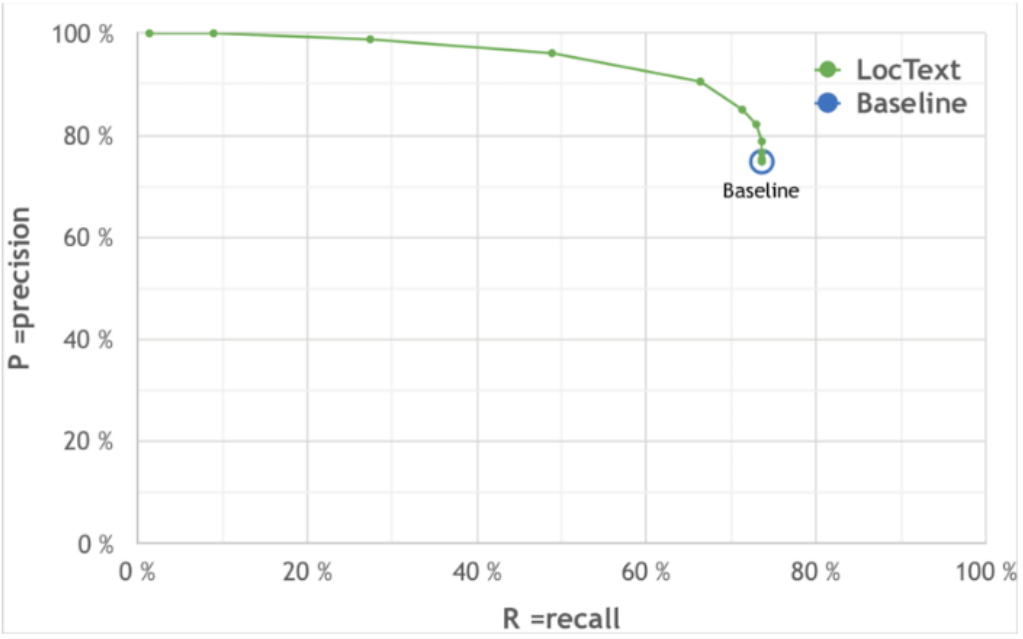


FIGURE 2.2: An example of a PR curve analysis using manually-annotated entities between LocText vs. Baseline [Cejuela et al., 2018].

2.8 OVERVIEW

2.8.1 TENSORFLOW API

TensorFlow (TF) is an open-source high-level ML library developed by Google in 2015. Its flexible numerical architecture facilitates simplicity especially, in DL tasks through the Python programming interface. TF and its high-level API called *tf.estimator* will be used as a framework in simplifying the machine learning (ML) algorithm for predicting gender in this research.

With regards to gender prediction, the TF API was used to build a supervised ML model that predicts gender from tweets relating to gender-based violence using tweet properties which comprises of text and user's description. Essentially, the DNNClassifier was used to build a deep feed-forward neural network which was trained on a labeled dataset so as to perform classification and prediction on an out-of-sample set.

The TF framework in building our model for gender prediction from labeled tweets on gender classification is benchmarked on the concept of a paper by Cheng et al. [2017]. They expanded on the different components of TF, like Layers, Estimators and Pre-made Estimators.

Under-the-hood, the TF Estimator is a versatile API for implementing ML algorithms [Buitinck et al., 2013]. It operates within the same functionality as scikit-learn [Pedregosa et al., 2011], in that the train-evaluate-predict loop is similar in interface. It is a framework that greatly simplifies ML tasks through the creation of computational graphs, specifying variables to assign, model training, serving and export into checkpoints repository for productionisation.

It works by offering APIs at different levels of abstraction, therefore, making common model architectures available for use from out-of-the-box while providing a library of utilities designed to accelerate experimentation, hence balancing competing demands for flexibility and simplicity. Moreover, the Estimator contains a model built with completion and encapsulates the following steps in ML:

- **Training**: The model is fit to a training dataset and determines hyperparameters.
- **Evaluation**: The fitted model is assessed on a separate dataset that it has not seen to check the goodness of fit.
- **Prediction**: The model is used to predict outputs for new inputs and implements inference.
- **Export for Serving**: This performs the *export_savemodel* method, a "a serialisation format which allows the model to be used in TensorFlow Serving, a prebuilt production server for TF models" [Cheng et al., 2017].

The concepts of *feature columns* and *input functions* are critical variables attributed to the Estimator's functionality. We propose to make substantial use of the pre-made estimator as a subclass of the Estimator. This greatly simplifies the implementation and programming of ML models. Pre-made estimators are parameterised not only by traditional hyperparameters, but also by using *feature columns*, a computing function that defines how input data can be interpreted, making out-of-the-box models flexible and usable across a wide range of cases. The framework described in Cheng et al. [2017] is implemented at the core of TensorFlow[5] and made available as a generalised [Dean et al., 2012] data flow graph.

Figure 2.3 shows the application of TF framework to ease the deep learning approach for data abstraction on gender prediction.

[5]https:/www.tensorflow.org/

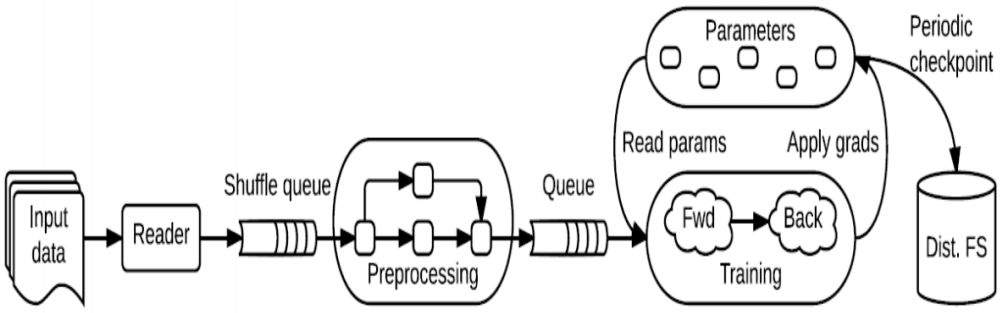


FIGURE 2.3: A "schematic TensorFlow dataflow graph for a training pipeline, containing subgraphs for reading input data, preprocessing, training, and checkpointing state" [Abadi et al., 2016].

2.8.2 TENSORFLOW SERVING

TensorFlow Serving[6] (TF-Serving) serves ML models into a production environment from TF as an API. It enables new algorithms and experiments to be deployed easily while maintaining the same built-in architecture. TF-Serving offers integration with TF models, but can easily be scaled and integrated on other models. It addresses the inference aspect of machine learning (ML), takes model after training and manages their operations by providing clients with versioned access on a high-performance, reference-counted loop. TF-Serving comprehensively supports TF models (naturally) with all its internals, but manages arbitrary versioned items (servables) with pass-through to their native APIs at its core. In addition, to train TF models, you can also use other inference assets, such as embedding, vocabulary and functional transformation configurations or even non-TF ML models. The architecture has a high degree of modularity. Some parts can be used individually (e.g. Batch scheduling) or all the parts can be combined. The running of an out-of-the-box TF-Serving supports:

Serving of TF model on a production environment for extended use.

Scanning saved files on local machine and export of TF models.

We extensively employed TF-Serving so as to push our trained model into the web by the injection of a custom-made `Node.js` server called the `Indexer` which is primarily designed for this research as a servable object. It scans the local file system periodically, loading and unloading models based on file system status and model versioning policy. This makes it easy to deploy trained models by copying the exported models to the specified file path while TF-Serving continues to run as shown in Figure 2.4

[6] https://www.tensorflow.org/tfx/guide/serving

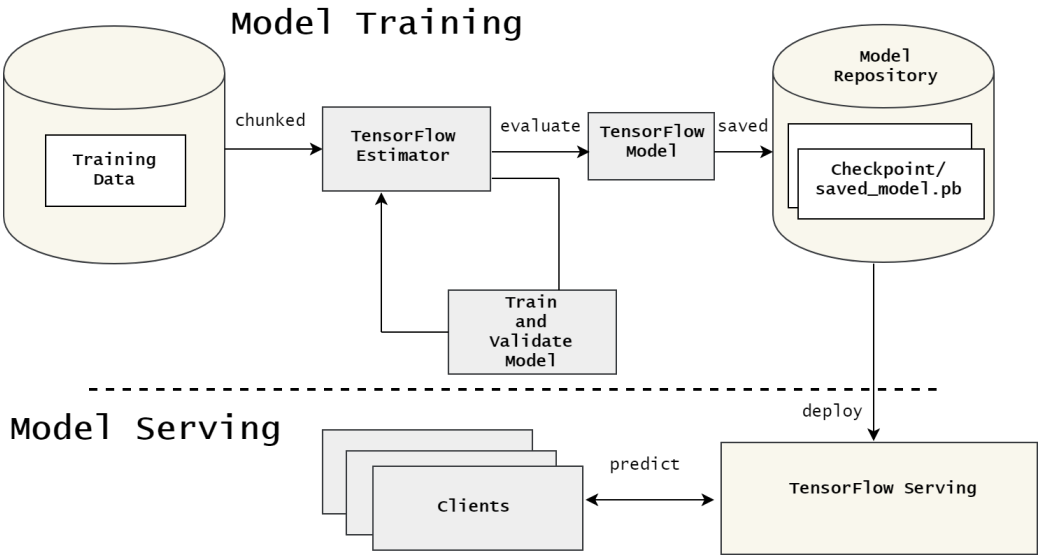


FIGURE 2.4: TensorFlow Serving Architecture

In the same vein, we initiated TF-Serving on a server called an `Indexer` pre-built of `Node.js` and its utility libraries. This was utilised for discovery and distribution of traffic in front of the instances. We equally introduced a communication protocol as an API in an instance of a gRPC[7] model server for response call and binary initiation of streaming tweets with a Predict API within the **Indexer**. gRPC is an open, high-performance remote calling process (RPC) framework running on HTTP/2 from Google. A few interesting improvements with HTTP/2 compared to HTTP/1.1 include support for multiplexing requests, bi-directional streaming and binary instead of textual transport. The `Node.js` script defines the commmunication path in serving a trained model in production, for example, the web. The algorithm that shows how this works is showed in Appendix A, B, C and D.

2.9 ELASTICSEARCH AND KIBANA: STORAGE AND ANALYSIS

2.9.1 USE CASE 1:

We reviewed `Kibana` and found that it has substantially grown since its inception in 2007 [Marquarding, 2018]. As an open source and a cross-platform, analytic and visualisation plugin for Elasticsearch (`ES`), its use and capabilities for visualisation of data and logs are dependent on the documents indexed on an `ES` cluster. Large volumes of data sets and log files can be reported, streamed, analysed and visualised from a tool like Kibana which is built on top of `ES`. Quite a number of scholarly studies have been put forward using the `ES` and `Kibana` tools. One such relevant

[7] https://grpc.io/

study by Lahmadi et al. [2015] was titled "A Platform for the Analysis and Visualisation of Network Flow Data of Android Environments". This study aimed to develop a monitoring system with functionalities in the collection, storage, analysis and visualisation of logs and mobile application network traffic. The data through the logs are then made available inside the ES engine and intuitively visualised using the Kibana platform. Its intuitive, browser-based user interface facilitates the dynamic creations of dashboards that reflect continuous changes of ES queries in real-time.

2.9.1.1 ARCHITECTURAL DESCRIPTION

A baseline for this study, which is presented in Figure 2.5, was to obtain and analyse network responses of smart devices through the collection of information in reference to traffic ingress and egress. It was discovered from the study that the network flow traffic referenced to the running application was measured on devices running on Android environment utilising the *Flowoid* application as the base monitoring probe. *Flowoid* enabled the recording and transfer of collected data using scalable architecture which enabled the flow data to be collected in a NOSQL database, indexed, visualised and finally analysed. The use of *device probes* for mainly the collection of flow data about running applications was the NetFLow for Android devices. This is the first probe and it is the *Flowoid* which is composed of a backend IP retrieving mechanism for network flow retrieval based on the geolocation of the data. The second probe is in the description and application of the log data running on each application. The data retrieval is achieved through the integration of "dumpsys" and "logcat" tools implemented by the Android environment. This functionality is expressed through the periodic collection of data using the "syslog" protocol, before it is parsed and stored.

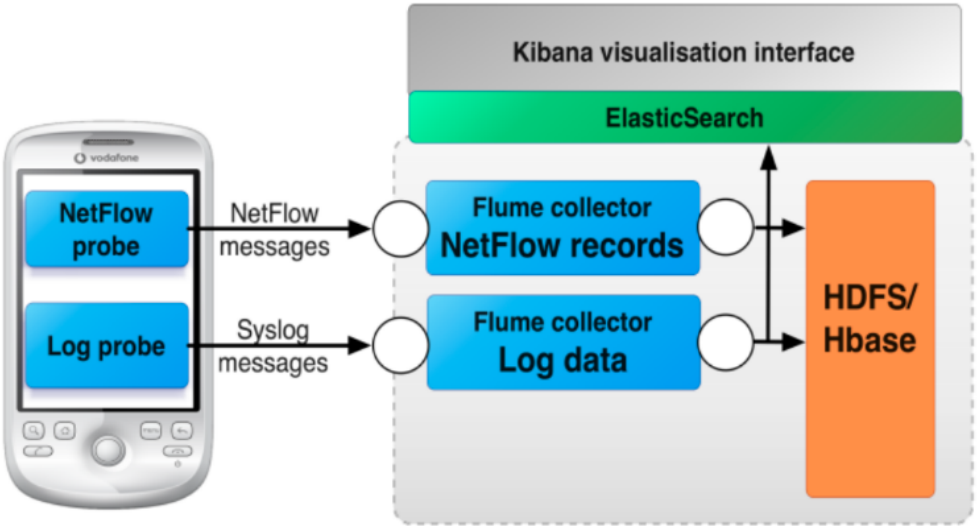


FIGURE 2.5: Architectural description of collected and analysed flow information [Lahmadi et al., 2015].

2.9.1.2 DATA EXTRACTION AND STORE

The extraction of data is facilitated through Apache Flume[8] for Flume based extractor, which is done with the help of interceptors. Each unit of data extracted is taken to be an event trigger which flows through the pipeline of a "Flume agent". It has the intelligence to recalibrate or drop event triggers in-flight, navigates through the source of a channel to a Sink and then to a centralised store. The JSON documents are initiated through the development of interceptors for NetFlow and "Syslog". The study was robust in storing up the JSON documents in a Hbase table.

2.9.1.3 DATA SEARCH AND VISUALISATION

This study by Marquarding [2018] reveals that data *search and visualisation* can be achieved from the combination of ES (whose role was to index data for ease of query and flow) and Kibana serves as a visualisation and aesthetic tool for developing a dashboard. Kibana interfaces with `ES` to facilitate search prompts, data aggregation and visualisation through the usage of different visualisation tools like *piecharts*, *tables*, *barcharts* and *tile maps*. Figure 2.6 presents an instance of multiple-layered piechart generated through `Kibana`.

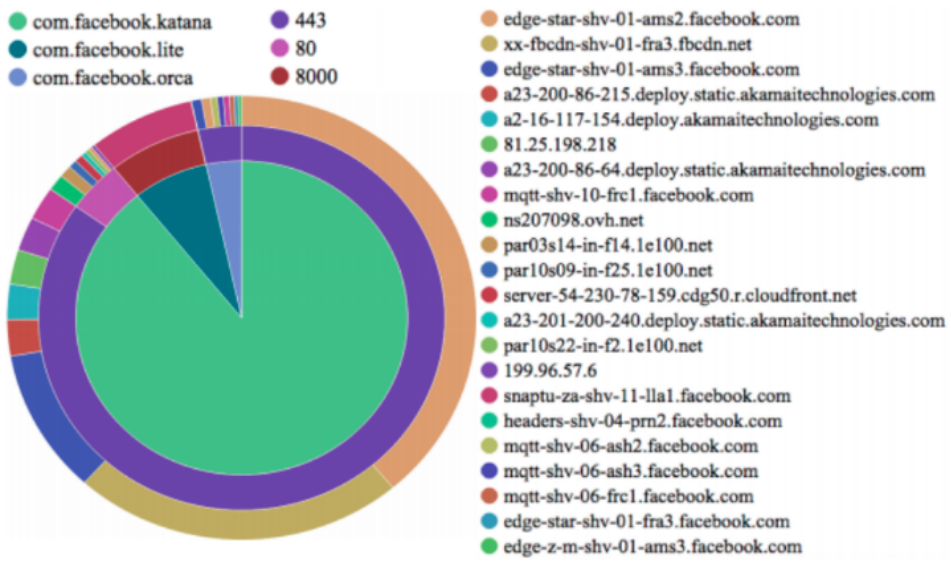


FIGURE 2.6: A piechart showing patterns of Android environment on a Facebook application [Lahmadi et al., 2015].

In summary, the interactivity of the monitoring system as shown in Figure 2.5 explores and identifies communications pipelines of varied mobile applications in an Android environment. This

[8] http://flume.apache.org/

study is significant to ours in approach through the provisioned services of ES and Kibana, but vastly different in regard to case and implementation.

2.9.2 USE CASE 2:

Over the years, the evolution of Internet-of-Things (IoT) technologies has gained its shared relevance of interest, and the vulnerability of IoT technologies have been evenly verified. A review of a study titled "Portable DoS Test Tool for IoT Devices" by Nagara et al. [2017] offers an unprecedented opportunity at analysing and reporting attacks performed on a target device before the display of captured traffic through a monitor for transmission of processed information. The study presents an attacker carrying out a denial of service (DoS) attack on a target device and discovered reports are transmitted to a monitor. When a ping request is initiated to the target device and the attacker pushes threatening information, ES and Kibana visualise the ingested information on the monitor in real time. ES is a full-text search engine with its dual functionalities as a data repository and an analytic tool. Kibana, which serves as a plugin for ES, is a data visualisation tool that displays log data on an internet browser. The study presented the attacker to be an embedded board odroid-c2 (ubuntu 16.04 LTS) of Hardkernel Incorporated in South Korea and described the monitor to be a MacBook Pro (macOS High Sierra) single-board with a high performing processor.

The responsiveness of the DoS attack by way of its effects were validated through these stated models. First, the study shows how packets under attack were collected through the mirroring function and investigated the response from the domain using a packet capture tool. Further to that, the state of the IoT technology to be attacked was investigated. It was also observed that both situations have been integrated in ES and Kibana and resolved by a load visualisation mechanism. Specifically, the study revealed how pings were visualised with regards to response time, traffic description and the digital representation of the target devices.

The snippet in Figure 2.7 demonstrates the appearance of attacks on static embedded system prototypes. It was understood that this prototype was responsible for the video captured on the server by the camera.

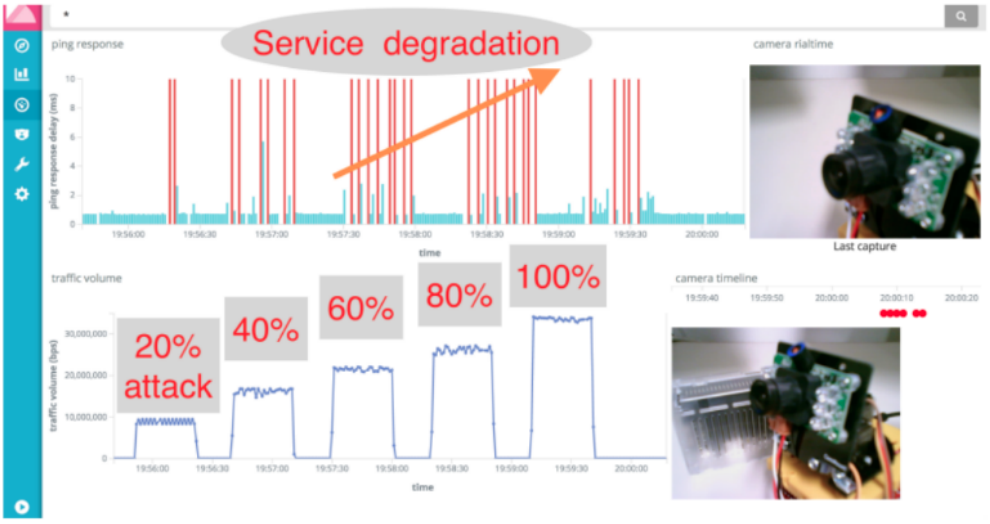


FIGURE 2.7: An embedded system showing levels of attack [Nagara et al., 2017].

In summary, we understood that the tool sufficiently checked to what extent IoT technologies were resistant or susceptible to DoS threats and attacks. Figure 2.7 shows the visualisation snippet of the attack on `Kibana's` aesthetic dashboard.

2.9.3 USE CASE 3:

Big data is somewhat vague but yet directional in the event a use case that aligns with its intended purpose and impact is efficiently defined. We consider another study that addresses the overall difficulty in log management at CERN [Hamilton et al., 2018]. An overview of this scholarly study which is titled "SCADA Statistics Monitoring Using The Elastic Stack" was conducted by Hamilton et al. [2018] to influence the ease of statistical access and error logs management through a more centralised design in a web application. The study presented 200 control systems under the management of the "Industrial Controls and Safety systems group at CERN". These control systems with a history of its large amount of devices, archive reports and changes in alarms to an in-house High-Performance Oracle database. The size of data is extracted from these thousands of devices and the inability of the Oracle database to handle this increase resulted in the challenge to easily view logs, analyse or utilise this data for a more advanced statistical model. The study highlighted two problems:

"It is not possible to easily obtain, for example, a list of devices that are archiving an excessive number of value changes. This is important to detect as it could indicate a faulty or misconfigured device."

"The controls applications also generate alarms and it should be possible to easily obtain information on the number of alarms from each application and device."

The log file servers *(WinCC OA Logs)* were overwhelmed by its continuous ingestion of log errors, prompts and information from logs of local files on the servers which these applications run. It was found that the examination of these logs was unavoidably difficult as they were distributed across multiple servers. The study made us understand that extracting and externally examining these logs on a centralised base without inflicting pressure on the production system like the centralised server was thoughtful.

The "SCADA statistics monitoring services" was built through the implementation of the Elastic Stack Suite[9]. This is further broken down into Elasticsearch, Logstash, `Kibana` and Filebeat. The inclusion of Filebeat was to serve the role of 'data shipper'. It takes text-based log files and sends them by reason to either `ES` or Logstash. Logstash is said to be an open-source "data processing pipeline" that is open to the ingestion of data from multiple sources. It transforms and channels data to various agents, one of which is `ES` which facilitate Logstash.

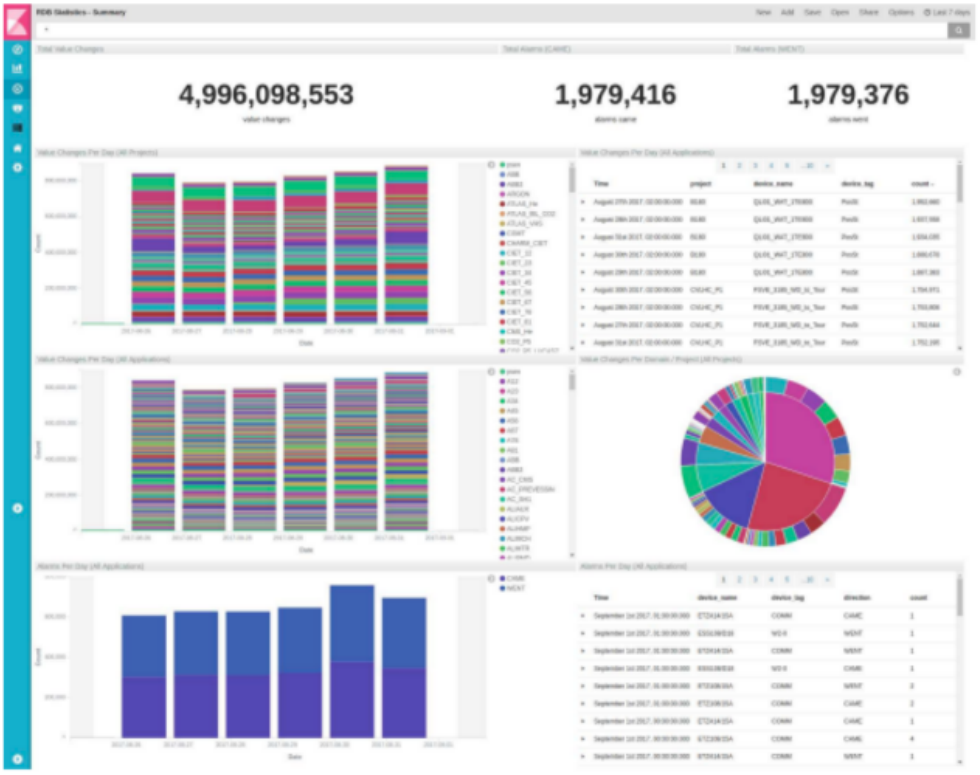


FIGURE 2.8: Dashboard from value change alarms data [Hamilton et al., 2018].

[9]https://www.elastic.co/products

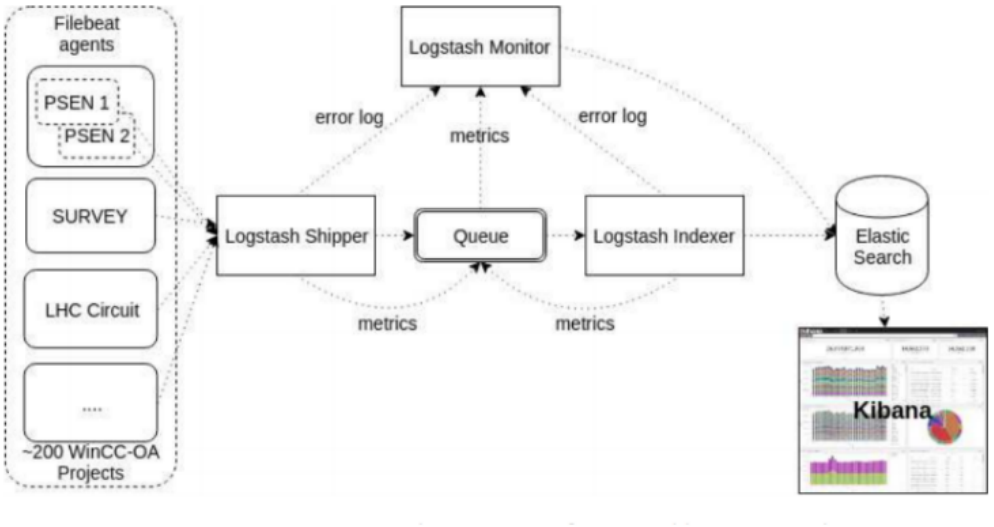


FIGURE 2.9: Collection of information through Logstash[Hamilton et al., 2018].

Figure 2.8 is an instance of the Kibana dashboard. `Kibana` is an `ES` plugin and a web-based visualisation tool which provides flexibility of real-time data visualisation in a variety of graphs, charts and tables.

The implementation of the Elastic Stack Suite for this study reproduced a web-based servicing system which allows for users to evaluate, visualise and query data; and to interface with application logs on the web system. The study tried to solve the problem of gathering information that addresses value changes and alarms through carrying out queries on a daily basis in order to gather information as shown in Figure 2.9 for all applications running in the server.

The study reveals that each JSON document in Elasticsearch comprises the estimation of "value changes" or "alarms", and "metadata" in description of "domain", "application" and "device names". This allows for the visual experience in `Kibana` at the "domain", "application" or the device level.

This study further underscores the varied applications of the Elastic Stack Suite. The development of the provisioned services in log management of devices at "CERN IT department" was through the OpenStack. It enabled the seamless set up which includes addition or removal of Logstash indexers. It is understood that the "Industrial controls application at CERN" facilitates the production of large volumes of data in a variety of domains. Hence, the need of an automated schema-free search and analysis engine, which is an instance of `ES` with its indices visualised in `Kibana`.

In summary, the study shows the potentials of being extended beyond its current use case as it reveals the high-level approach with which log files from various device applications at "CERN IT department". Thus, this study shows the significance of Kibana in helping with visualising outputs for further analysis.

2.10 SUMMARY

The broad area of the current research is on inferring gender as a demographic attribute and sentiment analysis from sexual violence-related tweets within South Africa. This research will make substantial use of deep learning neural networks and Google's TensorFlow to predict gender from an unlabeled dataset on gender classification. Our interest dwells on efficient analysis of Twitter data and how it can expose information that was previously unknown about sexual violence cases. This will be made possible through designing a web-based application which can help us visualise patterns on sexual violence for further analysis by stakeholders. The classification metrics were inclusive in this research to effectively evaluate the model. The next chapter presents the research methodology.

CHAPTER 3

RESEARCH APPROACH AND METHODS

3.1 INTRODUCTION

The literature review was presented in Chapter 2. It provided an overview of the technology used for gender prediction and sentiment analysis of tweet data as well as the reference of related works. Therefore, in this chapter (Section 3.4), we present the methods which includes the setup of the activity design and implementation. This section also describes the datasets used before the summary is presented in Section 3.5. However, the assumptions upon which this research is predicated are first highlighted in Section 3.2.

3.2 ASSUMPTIONS

1. The dataset from Kaggle (open-source data platform) was collected and serves as the ground truth data. This was used in validating the performance experience from our model on gender prediction.

2. An online community like Twitter permits registered users to be at liberty in generating any name of their choice. For instance, a female user may prefer using a male's name and a male may, in turn, prefer using a female's name. Beyond their self-declared user names and identity, no effort was considered in verifying names.

3. The dataset reflects real-time sexual violence tweets from registered Twitter users from South Africa and may be inefficient if a user changed location.

4. The corpus represents tweets in British English language only and not any other language.

5. Twitter is liberal to allow its users liberty at using screen names which could showcase alpha-numeric attributes like numbers and special characters. For example, £76r, 123-gh% and ijk!*. These are potential screen names which are not considered.

6. Because of Twitter's restrictions to the numbers of possible characters embedded in a tweet, users sometimes shrink content from websites in a URL to avoid violating Twitter's policy on character restrictions. We disregard for the purpose of this research all URL and links to web resources.

7. Twitter's Hashtag (i.e., #) is an intuitive and dynamic tagging metadata in coordinating topics. We only extracted sexual violence tweets verifiable through keywords and search tokens on rape, sexual abuse, sexual assault, sexual harassment and sexual violence. Any hashtag which forms a word away from the boundaries of sexual violence is not considered.

8. Within the aims and objectives of this research, it was only possible for us to infer gender and analyse sentiments from sexual violence tweets.

9. We clarify that a tweet text or data is appropriate to mean sexual violence tweet within the context of usage in this research.

3.3 CORE FRAMEWORKS

The gender predictor and sentiment analyser are custom made applications that represent the crux of this study. The significance of these applications described in Figures 3.1 and 3.2 are frameworks that drive the essence of this research in subsequent subsections.

3.3.1 GENDER PREDICTOR

We used the techniques of backpropagation as a deep learning algorithm to build an artificial intelligence (AI) model using DNNs with Google's TensorFlow to classify and predict gender from tweets. The AI model is able to discriminate between gender classes after training with labeled datasets. The label could either be male or female as indicated in Figure 3.1. The *user description* and *text* of a tweet object were chosen because it exposes the writing style of the author.

In the context of a social conversation, the aim of the AI model was to discriminate between the writing styles of both male and female as well as to predict gender given new unlabeled data [Imuede

et al., 2020]. This *out-of-sample* data is the real-time streaming of sexual violence tweets that is visualised in `Kibana`. To achieve this, the model was trained on the *bio - user description*, and *text* fields. After the training and testing, accuracies in both instances were achieved. This determined how well the model would perform before it was served on the server through TensorFlow Serving. We emphasize that the development of the model is not entirely focused on accuracy, but on how to use TensorFlow Hub[1] (TF-Hub) layers in a model for text classification. Nevertheless, accuracy has a potential of affecting how the model may perform against other similar models.

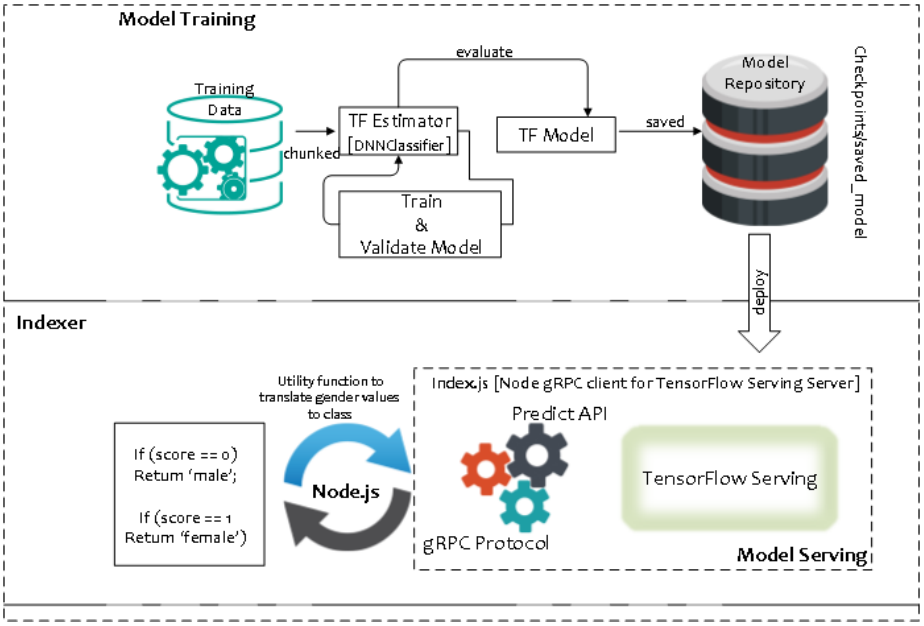


FIGURE 3.1: The framework for gender predictor

For more information on how the above framework ties up from the perspective of a TF high-level API, refer to Section 2.8 of Chapter 2.

3.3.2 SENTIMENT ANALYSER

The overall contextual clarity of a tweet document informed our choice to develop an application called Sentiment Analyser to help understand sentiment analysis within the context of this research. The idea from this development will enable us to identify sexual violence tweets as an actual instance of sexual violence expressed by a user. Furthermore, It gives us an idea on the measure of disparity expressed by a male and a female on a tweet related to `GBV`. It runs on the `Node.js` runtime and was developed primarily in JavaScript for this research. It answers the question on sentiment analysis in Chapter 1. We built the application using the current version of the AFINN word list which contained more than 3,300+ words with a polarity score associated with each word

[1] https://www.tensorflow.org/hub/

[Nielsen, 2011a]. This word list contains English language words which have been rated for valence. The valence of the words in the tweet is retrieved through the JavaScript framework from the list and is summed up. As indicated in Figure 3.2, the final score determines the sentiment and a negative score means the text has negative sentiment and vice versa. This is different from the lexicon-based approach implemented for the extraction of sentiment from text document by Taboada et al. [2011]. They used a framework called "The Semantic Orientation CALculator (SO-CAL)", which applies to the task of polarity classification implemented from word dictionaries. The idea of assigning a positive or negative label to a text is also centered around this.

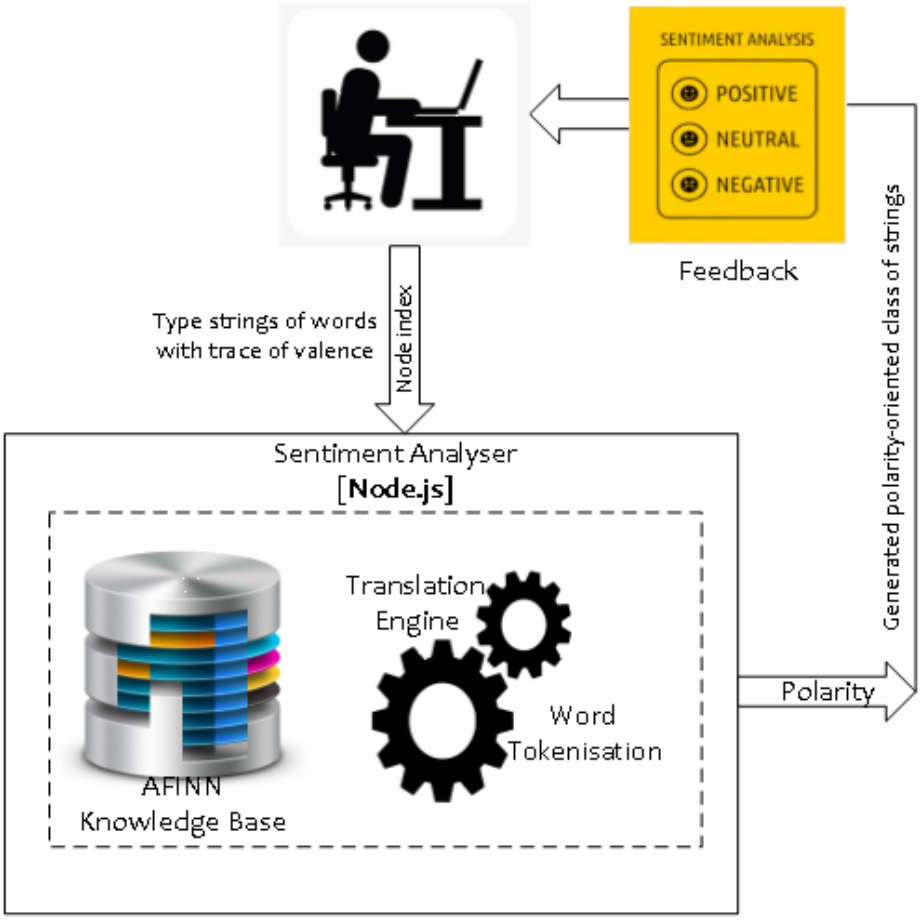


FIGURE 3.2: The framework for the sentiment analyser

3.4 RESEARCH METHODS & IMPLEMENTATION

With regard to the tools and approaches discussed in the literature of this research, our work in this area of study has proven to be novel because it attempts to investigate the disparity in reported cases of sexual violence in terms of gender representation and sentiment expressed. This is achieved through an interactive Web visualisation system for streaming live tweets on gender-based violence

(GBV) to infer hidden attributes from users' tweets. Figure 3.3 summarises our approach and further expand on the core frameworks earlier described in Section 3.3.

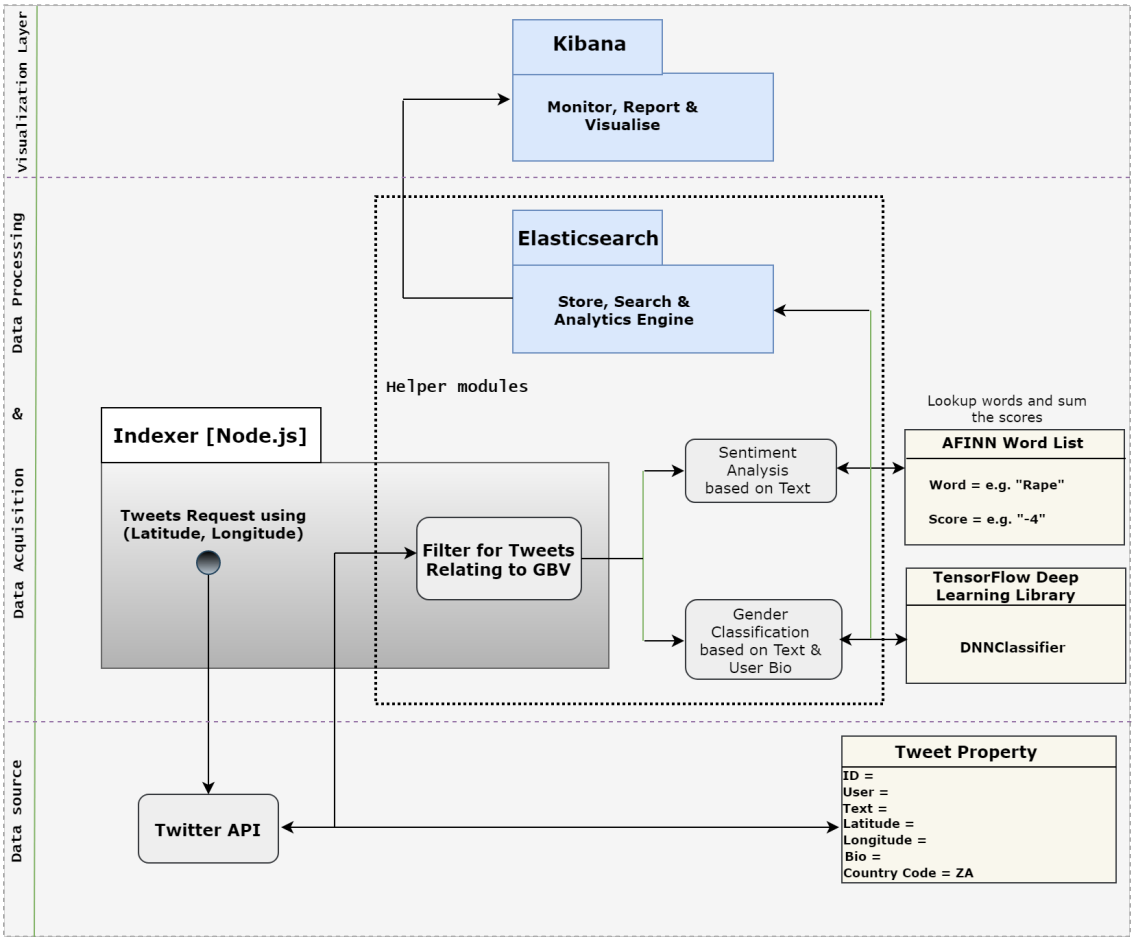


FIGURE 3.3: The overall system pipeline

Figure 3.3 is actually an activity diagram and in some sense provides the steps taken to achieve our goal from both a process and architecture perspective. Essentially, the system (a combination of integrated components) works in near real-time and continuously follows the processes stated below:

1. The `Indexer` running on `Node.js` will request a tweet from the Twitter API to provide all tweets from South Africa that contain particular keywords relating to `GBV`.

2. This data is retrieved by the `Indexer` and is validated to ensure that it is a tweet and is related to sexual violence, a form of `GBV`.

3. Valid tweets are then passed to the sentiment analyser and Gender predictor, which use the AFINN 111 scoring and the DFFNN methods, respectively to determine the sentiment and gender separately.

4. Both the determined sentiment and the predicted gender are then pushed into the `ES` module and are continually visualised using `Kibana`.

3.4.1 INDEXER

The `Indexer` is a `Node.js` Server-side Application whose role is to connect to Twitter's streaming API and ingest tweets in real-time from Twitter. At a high-level, the system will request data from Twitter API and then the Twitter API will respond with requested data to the system following the client-server architecture model shown in Figure 3.4.

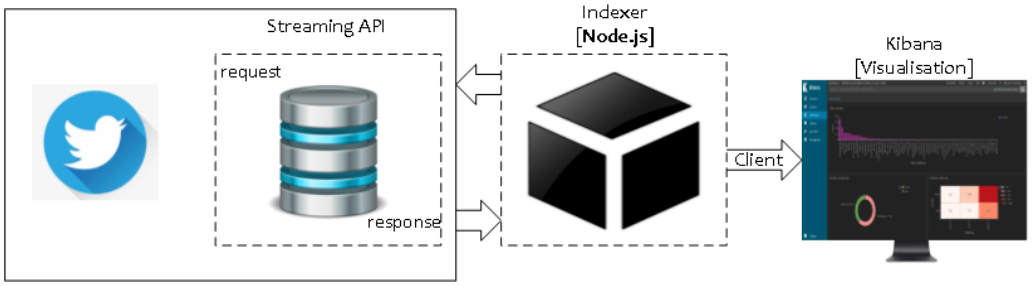


FIGURE 3.4: The client-server architecture "HIGH-LEVEL"

The `Indexer` pre-processes and annotates real-time sexual violence tweets to derive sentiments and gender through *user description* and *text* (tweet). In form of a "black box", it does the text analysis by revealing sentiments of the tweet and gender of the author. It primarily consists of two components; namely, the Sentiment analyser and the Gender predictor described in the subsections below and introduced in Chapter 1 of Figure 1.1.

3.4.2 CODE DESIGN FOR GENDER PREDICTOR

We specified *input_functions* by interface, and not by inheritance. To facilitate the use of external objects with TensorFlow, inheritance is not implemented; code conventions instead provide a coherent interface. The hub of the object is TF Estimator — *tf.estimator*, which implements a fit method, accepting an input data array as arguments and, optionally, an array of labels for the supervised task. We used a *predict* method which can be implemented by supervised estimators, such as DNNClassifiers. Some estimators, for instance, principal component analysis (PCA), which we called transformers, introduce a technique of transformation, returning modified input data. Estimators can also provide a *score* technique that is an increased evaluation of the goodness of fit method: a log-likelihood, or a negated loss function. Another significant item is the confusion matrix on both splits of train and test sets. In Chapter 2, we discussed the estimator as a TF high-level API that deals with the creation of computational graphs, initialising variables, training the model

and saving checkpoint/logging files to be visualised in TensorBoard. The two fundamental concepts of the estimator upon which every other internal depends are the **feature columns** and **input functions**.

For training and evaluation, input functions are used to pass input data to the model. Feature columns are specifications for the interpretation of the input data by the model. The motive is that we want to predict the gender from collected tweets through the features (user bio and tweets) in the datasets. The rationale behind this was to develop a ML algorithm through the implementation of a DNNClassifier that is intelligent enough to represent our features in different ways using the feature columns. This was achieved from building an input function which will push data to the estimator. Feature columns connect the input data to the training functionality embedded in the estimators and evaluates the model.

Essentially, we trained a DNNClassifier model with 2 hidden layers. The DNN is a 3-layered neural network (NN) with two hidden layers of 500, 150 neurons respectively and an output layer with 3 neurons which represents the different gender classes (male and female) and brand (non-individual). The links (weights) are the connections or synapses between neurons across different layers and edges.

Figure 3.5 shows the features, hidden layers, and predictions (not all nodes in the hidden layers are completely represented).

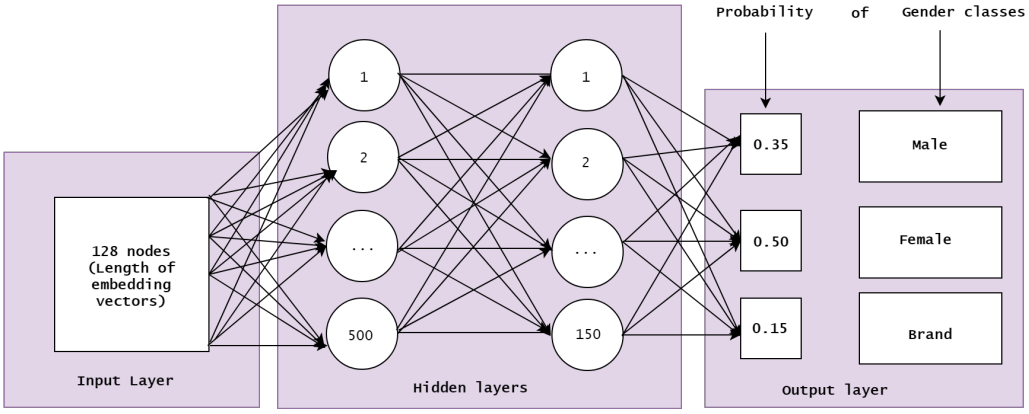


FIGURE 3.5: The architecture for gender prediction (not all nodes in the hidden layers are represented)

The general workflow is presented in form of a pseudocode which is a high-level overview on the systemic sequence detailing steps in designing our model for gender classification and prediction.

Algorithm 1 Pseudo-Code

Step 0 Import all dependencies.

Step 1 Remove all common stopwords (and, the,...).

Step 2 Clean text by removing symbols (hyphens, apostrophes,...) and spaces.

Step 3 Load tweet data from file, split data into training and test sets, assign numerical values to gender categories and combine tweet and user description fields.

Step 4 Build function to hold training input data.

Step 5 Build function to hold sample prediction training input data.

Step 6 Build function to hold testing input data.

Step 7 Build a function for testing a single example.

Step 8 Transform the all_features field into an embedding vector.

Step 9 Build the estimator which represent the neural network.

Step 10 Invoke the neural network passing in the input function built earlier.

Step 11 Evaluate neural network by inputing the testing input data.

Step 12 Use the neural network to predict a single input.

Step 13 Print evaluation results.

Step 14 Save model into machine.

Step 15 Visualise the model and evaluate the loss metrics using TensorBoard.

3.4.3 DATA OVERVIEW

The dataset used in training the gender predictor model was retrieved from Kaggle through the "Data for Everyone Library On Crowdflower" and is always available for the community, free of charge. Contributors to the project were asked to mainly analyse a Twitter profile and determine if the user was a male, a female or a brand (non-individual) [KaggleGenderClassification]. The dataset presents 20,000 rows and columns, each with a username, a random tweet, account profile with image, location, link and sidebar colour. However, we mainly focused our attention on the subset of the features (tweet and profile description).

The tweet and user description of a Twitter user were the attributes chosen because it exposes the stylistic convention of the author. We loaded and read the data in a comma-separated values (CSV) format through **pandas**. This is an automatic and explicit data alignment library for computational purposes in PandasDataframe. This revealed the default parameters of the datasets containing the index, the column headers and tweet itself. We extracted the components through a series of cleaning and preprocessing phases before it was finally fitted to the model. TF fits well with **pandas** and offers a variety of helpful functions to work with.

The model runs a DFFNN to determine the users gender. According to Sola and Sevilla [1997], depending on the activation function of the neurons, backpropagation networks require some pre-treatment of data used for training. So, we ensured possible noises like special characters, articles, spaces, irrelevant words were removed. The purpose for text normalisation was to obtain good result within a significantly reduced amount of time.

The basic reason we conducted data cleaning from human understandable language in tweets was to enable a machine-readable format (convert features to Tensors and fit to model). We started tweet normalisation to prepare our data for machine learning.

Below we list what tweet normalisation means for this research:

1. *Convert all letters to lower case.*
2. *Convert all numbers to words or delete numbers.*
3. *Delete punctuations marks and any other diacritics.*
4. *Delete white spaces.*
5. *Expand any form of abbreviation into a complete sense of thought.*
6. *Remove stop words, sparse words, and special characters.*
7. *Perform text canonicalisation.*

3.4.4 FEATURE ENGINEERING

3.4.4.1 ENCODING THE TARGET VARIABLE (GENDER)

The encoding of categorical variables is a vital step in the process of data manipulation. We have several approaches, but we have implemented the one-hot-encoding algorithm to convert text attributes into numerical values. The **pandas** package of Python provides several approaches that can be used to transform categorical data into appropriate numerical values. The basic strategy about one-hot-encoding is to map each category such as male, female and brand in our gender classification dataset to a new column and to assign the column a value of 0, 1 and 2.

3.4.4.2 WORD-EMBEDDINGS FOR FEATURE COLUMNS

Feature columns are described as objects that models use as raw input data from a feature dictionary. When a model is built using an Estimator's API, a list of feature columns is created that describes

each of the features you required by the model. The TF-Hub library provides a column of features that applies a module to the text function provided and further passes the module outputs. In this research, the *nnlm-en-dim128 module* is initiated. It is a module for Text embedding on feed-forward Neural-Net Language Modules with pre-built of out of vocabulary (OOV) API. OOV is used in a larger machine learning framework where strings of text are mapped to 128-dimensional embedding vectors. In using this module, we note the following important facts:

- The module requires as an input in a 1-D string tensor a batch of sentences and phrases.
- The module takes care of sentence preprocessing. For example, punctuation and space removal.
- The module runs on any input. For example, in 20,000 buckets, *nnlm-en-dim128* hashes words not in vocabulary.

In any case, deep learning systems are built to work with numerical values as input features. Therefore, tweets are represented in a way that reflects this constraint. In tackling this constraint in terms of text representation, we used word embedding as against the likes of bucketization, scaling, crossing features to convert tweets to Tensors of vector matrix before fitting to the model. Embeddings translate large sparse vectors into a lower-dimensional space that preserves semantic relationships.

3.4.4.3 TRAIN AND TEST SPLIT

We used the Estimator to split data for convenience. The Estimator provides a **train_test_split** function for splitting a **pandas** DataFrame into a training and testing sets. The function (**train_test_split**) takes in the **all_features** and the target (**gender**) as parameters and returns and encoded mapping of either **0** or **1**. 80% of the dataset was used for training and the rest for testing the model before deployment into production. The training split controls what portion of the data should be allocated to the train set; and it looks like this. **train_df, test_df = load_data_from_file()**.

3.4.4.4 BUILD INPUT FUNCTION

The model, through the Estimator, was trained to pass through the features and targets. The Estimator requires the implementation of an input function. The input function returns a tuple containing:

- A dictionary that contains the feature column type as class labels and maps them to their encoded variables.

The looks of an input function takes this structure. **train_input_fn = tf.estimator.inputs.pandas_input_fn**()

Essentially, TF provides functionality for feeding a **pandas** Dataframe straight into a TF-Estimator. The **tf.estimator.inputs.pandas_input_fn** has many parameters but we only have use for the following:

- The pd.DataFrame.from_dict (**X**) object has the features in **all_features**.
- The **tf.estimator.inputs.pandas_input_fn** is a function that is intelligent enough to exclude the label (**y**) from (**X**). **X** is simply specifying the column in **y** that should be used as a label.
- Batch_size: A number of hard-coded example specifying the batch size.
- Shuffle: A boolean initialiser to either shuffle or not.
- Epoch: A number of times to iterate over the entire dataset such that each in-sample data has been seen once.

In summary, we built an input functon to train, evaluate and predict data which were passed into the Estimator framework using **tf.estimator**. The **tf.estimator** serves input functions that wraps Panda dataframes.

3.4.4.5 INSTANTIATE THE MODEL

The gender prediction task is a classification problem. Luckily, within TF's library are several pre-made classifier Estimators. We concentrate on **tf.estimator.DNNClassifier** as our choice of API which is used for deep multi-class classification models. In Figure 3.6, We instantiated the DNNClassifier by passing the list containing the feature columns to the **all_features** parameter. We defined the **model_dir** parameter where TF will store the DNNClassifier's graph and checkpoints including other informations using the **estimator.export_savedmode** function. We were able to visualise the model graph (architecture) and loss metrics by pointing the saved model directory to TensorBoard. We made use of the **ProximalAdagradOptimize** optimizer.

```
# Builds the estimator which represent the neural network
estimator = tf.estimator.DNNClassifier(hidden_units=[500, 150],
    feature_columns=[embedded_text_feature_column],
    n_classes=3,
    model_dir="./model",
    optimizer=tf.train.ProximalAdagradOptimizer(learning_rate=0.06,
    l2_regularization_strength=0.004))
```

FIGURE 3.6: Code block to instantiate the model to choose between two classes

3.4.4.6 TRAIN THE MODEL

We trained the model by calling the trained method in the Estimator. This wraps the input function as the Estimator expects in Figure 3.7.

```
#train model

estimator.train(input_fn=train_input_fn, steps=5500)
```

FIGURE 3.7: Function to train the model

The **steps** argument instructs the method after a number of training steps to stop training.

3.4.4.7 EVALUATE THE MODEL

The snippet in Figure 3.7 represents a trained model which brings us to obtaining some performance statistics. The following code block in Figure 3.8 evaluates the accuracy of the model based on the test data.

```
# Evalute neural network by input the testing input data
test_eval_result = estimator.evaluate(input_fn=predict_test_input_fn)


# print evaluation results
print("Test set accuracy: {accuracy}".format(**test_eval_result))
```

FIGURE 3.8: Snippet for model evaluation

We stressed that we did not pass the *steps* argument to evaluate the model, unlike our call to the train method. The **eval_input_fn** yields only one data epoch. Results from the **eval_result** dictionary also includes the average loss (mean loss per a single sample), the loss (mean loss per mini-batch) and the global step estimator value (number of training iterations).

3.4.4.8 MODEL PREDICTION

We used the trained model to predict gender on the basis of unseen data. We make predictions with a single function call, as with training and evalution before saving the model.

```
# Use the neural network to predict a single input
predict_tensor = estimator.predict(input_fn=predict_single_input_fn())


for result in predict_tensor:
    print('\r\n\r\n class label: {} \r\n\r\n 0 = male, 1 = female'.format(int(result['class_ids'])))
```

FIGURE 3.9: Snippet showing model prediction

3.4.4.9 SAVED MODEL

TensorFlow saves variables to the tensor values in binary checkpoints that map variable names. The snippet in Figure 3.10 represents a single function call.

```
# save model
estimator.export_savedmodel("./model", serving_input_receiver_fn)
```

FIGURE 3.10: Function to save TF model

We used the supported format of SaveModel from TensorFlow format to have our predicted model saved.

3.4.4.10 VISUALISING LOSS AND MODEL GRAPH WITH TENSORBOARD

TensorBoard enables visual experience related to a trained model's internal architecture. If we want to visualise it, we merely need to set the **logdir** parameter to the folder where we saved our model. We can set the **model_dir** parameter to store the model when we instantiated the model. TF automatically saves output in the model's folder checkpoints and event files for visualisation with TensorBoard. We visualised the loss curve under the Scalars section on TensorBoard amongst other metrics variations like average loss for Estimators. We shall present these graphs in the result section in Chapter 4. However, Figure 3.11 is a complete representation of our model's "main graph" and "auxilliary nodes".

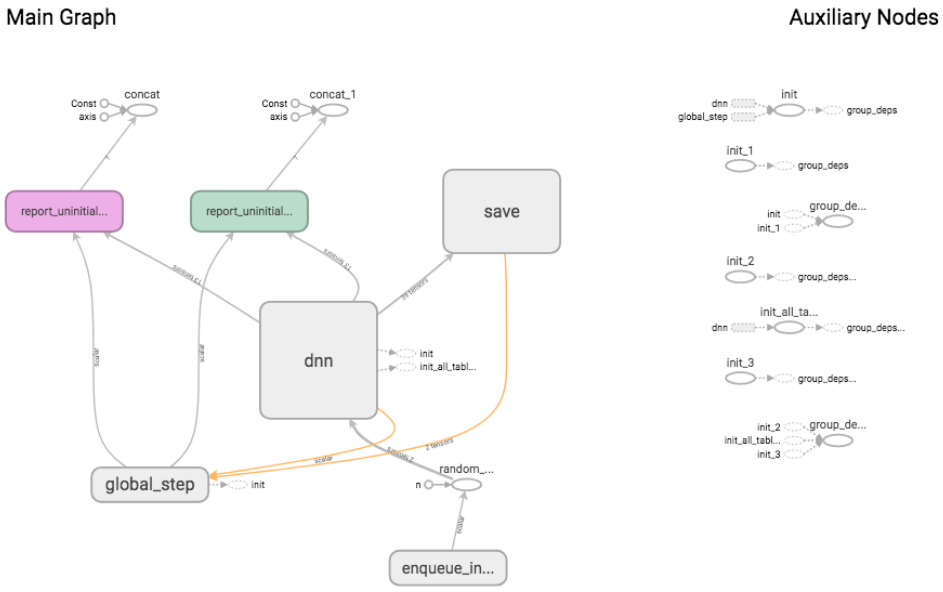


FIGURE 3.11: Model architecture showing computational graph

The above representation reflects the main components of the model. This can be explored through clicking on the nodes and subgraphs. We can check out different parts of the computational graphs which can help facilitate the creation of newer estimators for different task.

3.4.5 ELASTICSEARCH

The store, search and analysis of real-time sexual violence tweets is made possible by making use of `ES`. The tweet sent by the **Indexer** as a result of the connection implemented through the Twitter streaming API was indexed in `ES` as shown in Appendix B. It is based on an Apache lucene architecture; it allocates a multitenant-dependable search engine with a scheme-free JSON document. It is a reliable, free-source tool for efficient extraction, aggregation and distribution of large volumes of data from vastly different sources to a centralised data storage. Each unit of a tweet is annotated and indexed as an instance in `ES`. Through a JavaScript framework, the tweets were ingested by the `Indexer` and stored in `ES` with manipulation into a JSON document as preferred through `Kibana`.

3.4.6 KIBANA

At the phase of monitoring, reporting and visualisation of streaming sexual violence tweets, the output from Indexing on `ES` is visualised and analysed through an aesthetic dashboard creation in `Kibana`. `Kibana` is vital in this research because it reveals real-time sexual violence related tweets, which is engineered by connecting the streaming API to the `Indexer`.

3.5 SUMMARY

We summarised this chapter which touched on specific methods and tools that have been experimented. The idea of using `ES` and `Kibana` for storage and visualisation were introduced. The `Indexer` (the system's hub), which is developed using `Node.js` facilitates the connection to Twitter through the streaming API. This enables the streaming of sexual violence tweets from a predefined location. Finally, the knowledge of detecting which gender tweeted an instance of sexual violence and the sentiments were defined. The next chapter presented results which were derived from experiments conducted in relation to the stated approaches.

CHAPTER 4

EVALUATION AND RESULTS

4.1 INTRODUCTION

In the previous chapter, we presented the core components that drive the proposed research approach and methods through a deep learning construct. This chapter presents and provides the results retrieved from sequentially running the DNNClassifier through a parallel structure on TensorFlow and visualising the model's internals in terms of graphs on TensorBoard. It further showed the performance of the algorithm by evaluating it on a binary classification task such as gender.

4.2 DATA SET

In reference to the aggregated sample of data collected from Kaggle, we examined a large corpus of Twitter accounts (approximately 20,000 rows and columns) on gender classification to verify whether they belonged to a man or woman. We used historical tweets and profile description (the "about me," if you will), as features in training a machine learning model for gender prediction. The essence is that when you supply the *user description* and historical *text* (tweet) from a random Twitter account profile, the model will predict the gender class attributed to the profile. We clarify that the data collected from Kaggle is relevant in terms of the demographic distributions of male and female in relation to the Twitter social network, but not entirely relevant to our research direction in terms of gender-based violence `(GBV)` as none of the previously collected tweets were classified as

such. By way of emphasis, a South African contextual data is not available in this corpus, even if our focus is in South Africa. We only used the data on gender classification from Kaggle as a rationale in conceptualising and developing a predictive model whose resultant pipeline was indexed and channelled to fetch or collect tweets within our predefined boundaries on sexual violence.

By way of experimentally exploring the data, the collected set of data was analysed which showed the gender variables with female, male, brand and unknowns being counted respectively as 6700, 6194, 5942 and 1117.

Since we are particular about gender classification as “male” or “female”, the aforementioned gender variables were filtered to align with this objective by eliminating brand. From which, it was noticed that the distribution of males and females in the dataset was roughly equal which reflected a low chance of sampling bias. A problem such as class imbalance may have arose in this instance. This problem of unbalanced data refers to class instances being unequally distributed. By using the ”Under-sampling” technique, this imbalance was greatly reduced. By randomly removing samples from the majority class, this simple method enables balanced datasets to be generated which, in principle, lead to classifiers not biased towards a certain class [Vogel and Jiang, 2019]. Thus, to further mitigate this, the text variables were evaluated, and the data frame partitioned into training, validation and testing sets. Essentially, through the predictive algorithm, gender classification was conducted so as to automatically classify gender from streaming sexual violence tweets on the `Kibana` platform through the API of TensorFlow Serving. This is explained in Chapter 2. The machine learning convention used 60% of the data set for training, the validation set was 20%, and the remaining data (20%) was assigned for testing. Within the training portion of the split, the training set was further split and mapped as male (0) and female (1). We then feed the data as features into the model through defined functions.

Firstly, the deep learning algorithm looked at each data row (which in this case was a tweet and a profile description) and fitted the vector translation into the model. Then, through the usage of a transfer learning mechanism from Google’s TensorFlow library (TF-Hub), it looked for patterns by way of extractions. From the mapped training sets (male (0) and female (1)), the frequency with which words were used was defined as an identification index. Further, since reshaping, one-hot encoding (this is the binary encoding of the data field to get the target) and splitting were used in the algorithm, the amount of tweets that were most associated with men and women during testing for prediction on gender were determined.

After which, it was observed that females were significantly more likely to engage in topics on the Twitter network as compared to their males counterpart. According to Lasorsa [2012] and Burger et al. [2011], possible factors which were considered significant enough to have influenced the results as to variations in sampling interest were explored. These include factors such as availability of data, access to smartphones and infrastructure development like the internet in rural areas, etc.

Social norms are equally a point in case. Previously, females were expected to fit into prescribed roles which limited or completely eliminated their freedom of expression/speech. But it is not the case now. Currently, females tend to express felt injustices or grievances that threaten their collective rights as a female in almost all print and social media platforms [McGregor and Mourão, 2016].

4.2.1 RATIONALITY OF RESULTS

The validation set was introduced to our model's experiment because it allows for repeatable comparison of the varied parameters/architecture against the same training data and weights in the DNNClassifier. We used the split in the validation set for tuning the DNN's hyperparameters and visualise its effect on the predictive accuracy of the model. For clarity, the training set was used in building the model whereas the validation set on the other hand enabled fine-tuning of the parameters/architecture in a way to strengthen the model's predictive power measured on the basis of accuracy. Then, we introduced the test set to verify the DNNClassifier's predictive accuracy on new data not seen before by the model.

4.2.1.1 ANALYSING THE MODEL

By way of model selection, we critiqued our original fitted model by generating sub-models from tweaking the hidden layers of the model and learning rates to generate new models. We did this in order to be able to make an informed decision when selecting a model to serve for production through TF-Serving. We explored an advanced visualisation tool from TensorFlow called TensorBoard, a handy API that facilitated the internal inspection and debugging of our model's statistics. The original model was split into 3 sub-models with varied architecture, basically, from parameter tunning. Through our model structure and checkpoint files, we instantiated TensorBoard displaying the model's computational graph under the hood. By this, we were able to justify which model from the split was best to be suited for production.

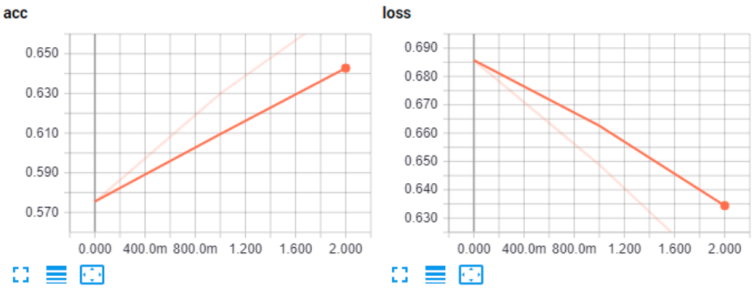


FIGURE 4.1: Graphs on model1 showing accuracy and loss



FIGURE 4.2: Graphs on model1 showing accuracy and loss on the validation set

Our model's performance from a predefined number of steps demonstrated how well the model behaved over time. Figures 4.1 and 4.2 display results from an initial run which represents outputs from splits on the train and test sets. The metrics displayed on the graph showed performance on unseen data. The experiments performed is so that we can arrive at a model that will generalise well on unseen data. For this, we changed a few parameters in the previous model for a new and better experience. To initiate this, we altered the architecture from the hidden layers through to the learning rates. Also, the distributive measures on the train/validation split and saved results on a different model log and visualised as shown in Figures 4.3 and 4.4.

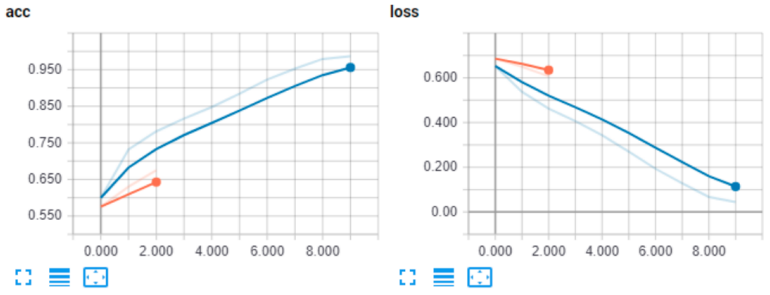


FIGURE 4.3: Graphs on model2 showing accuracy and loss

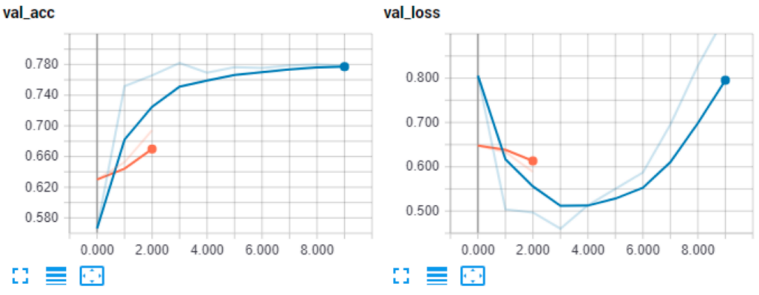


FIGURE 4.4: Graphs on model2 showing accuracy and loss on the validation set

We noticed the validation loss in Figures 4.3 and 4.4. Loss defines the measure of error [Bartlett et al., 2002]. The lower the loss, the better the model (unless the model has memorized the training

data), which in this case is a function of the classification algorithm. The loss is verified on the training and validation sets, and how well the model does for these two sets is its interpretation. From the graphs in Figure 4.4, the validation accuracy improves from the previous run. However, it showed signs of decreasing which would impact negatively on the validation loss. In this way, we say the model is possible to over-fit. We try to understand a possible cause of this and found out that the model is trying to decrease the in-sample loss. The best measure in this instance was to optimize the model otherwise the model will keep memorizing input data and the in-sample accuracy will increase, but the out-of-sample, and any unseen data when fed into the model, will perform poorly.

4.2.1.2 OPTIMIZING THE MODEL FOR SERVING

After we arrived at a model that worked, we optimised the model in Figure 4.5 and 4.6 through the modification of the nodes per hidden layer, the optimizer, learning rate and L1 to L2 regularisations. Essentially, we loosely toyed with the model's internals as we try to tinker around it. We took into thought the randomness in models and that no two rounds of optimisation will represent the same output, however close they might be identical. Through the creation of sub-models, we understand the need to increase the size on the computing resources. We installed TensorFlow-GPU so as to simultaneously run our combinations.

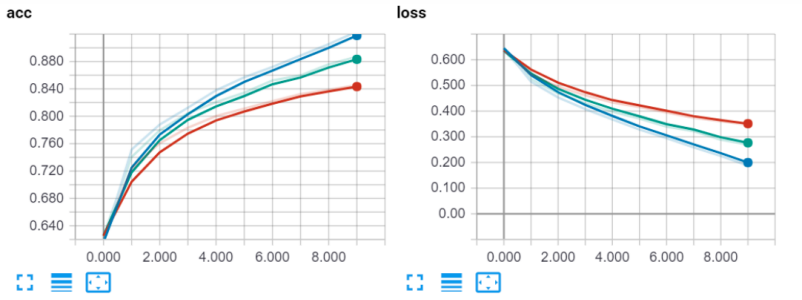


FIGURE 4.5: Graphs on model3 showing accuracy and loss



FIGURE 4.6: Graphs on model3 showing accuracy and loss on the validation sets

Results showed how tempting it is to select a model with the highest validation accuracy, but we instead tend to go for the model with the best (lowest) validation loss. We mentioned earlier that, when it comes to model selection, there are some randomness, however, trends should be noticed.

4.3 MODEL EVALUATION ON TEST DATA

As an interpretation of the DNNClassifier for the train/test split metrics, Tables 4.1 and 4.2 showed the variations in the results. The training accuracy which is founded on the knowledge base of the data for which the model is familiar is seen to be higher by 80% while the testing accuracy which is based on the data the model has not seen prior to training is lower by 68%. This points to the fact that our selected model performed well on the test set and represents a good performance measure of the model. An increase in the regularisation coefficient (L1 to L2) proved to improve this result as shown in Figure 4.8 and demonstrated in Appendix E.

TABLE 4.1: Train matrics after training using the DNNClassifier

	Accuracy	AUC	Precision	Recall	Loss
0	**0.8075**	**0.880475**	**0.781705**	**0.87259**	**0.38949**

TABLE 4.2: Test matrics after testing using the DNNClassifier

	Accuracy	AUC	Precision	Recall	Loss
0	**0.683455**	**0.7763**	**0.702957**	**0.778853**	**0.3492445**

As expected, re-running the model multiple times improves the learning rate and generates better accuracies on the training set with the epoch set to false. When the model learns more, it is an improvement on the accuracy which also impacts positively on the misclassification rate and suggests better performance evaluation on the basis of accuracy.

In comparison to results generated from Crowdflower's post[1] on experiments carried out on this same dataset, it was observed that the result was pretty lacking in details about what kind of model they used to predict Twitter user gender. The only information they showed was "we ran the tweets through our AI feature", and that they achieved about 60% accuracy on their three-way (male, female, brand/organization) classification task. But our quick run-through showed an improvement in algorithm and results presented.

[1] https://www.figure-eight.com/using-machine-learning-to-predict-gender/

4.3.1 CONFUSION MATRIX

An interesting statistical measure used in evaluating a model is the confusion matrix, particularly for a multi-class problem. It enables the percentage of properly and wrongly labeled instances to be visualised. In Figure 2.1, we can readily see how fairly biased our classifier is and whether it makes sense to distribute labels. Ideally, the biggest estimate of prediction is spread across the diagonal. We evaluated the metrics used from an imbalanced dataset through the knowledge base of performance measures for binary classification based on the test set. The confusion matrix, discussed in Chapter 2, as a measure of performance also revealed the misclassification of the algorithm. The performance of such an algorithm is commonly evaluated using the data contained in the matrix. Table 4.3 shows the confusion matrix for a gender class.

TABLE 4.3: This is an excerpt from our confusion matrix in Figure 2.1

X	Y	Classified As
x1	y2	X = Male
x2	y1	Y = Female

The variables in the confusion matrix of Table 4.3 have the following connotations in the context of this research:

1. **x1** is the number of correct predictions that an instance is negative.
2. **x2** is the number of incorrect predictions that an instance is positive.
3. **y1** is the number of correct prediction that an instance is positive.
4. **y2** is the number of incorrect predictions that an instance is negative.

We have trained a gender classification model to distinguish between a male tweet and a female tweet and implemented the confusion matrix to summarize the results of testing the algorithm for measure of correctness. The confusion matrix is evaluated based on both the testing and training sets and to understand the distribution of misclassifications. We visually examine these expectations in Figures 4.7 and 4.8.

The 60% split of the dataset for training the model reflected how confused the model resulted from classifying a male from a female in Figure 4.7

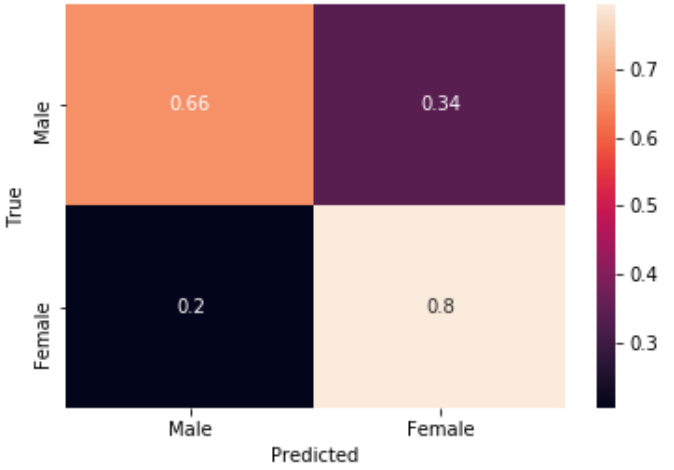


FIGURE 4.7: Confusion matrix on the training data

The variation is expected even in a real-world scenario where the way a male discussed an issue might be very similar to the way a female would discuss the same issue. It should not be interpreted that our model was biased in this very expected situation, but that the model actually did perform considerably well in such an instance.

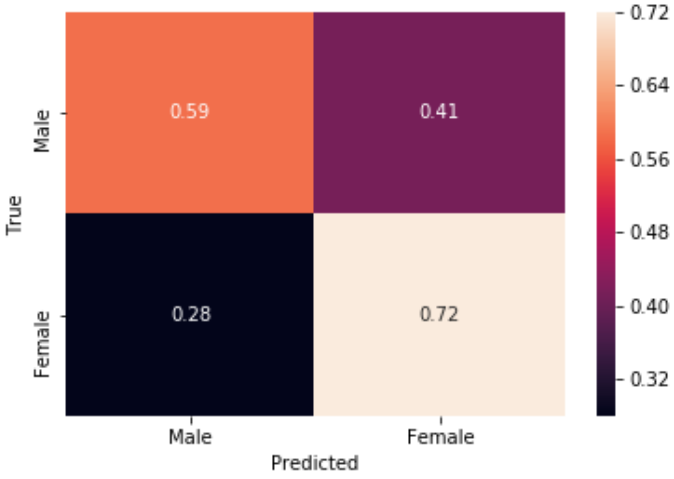


FIGURE 4.8: Confusion matrix on the testing data

The total sample for the test set is a 20% representation of the aggregated sets. The resulting confusion matrix is presented in Figure 4.8 after fitting into the algorithm. As expected, we point to infer that the algorithm had challenges in distinguising between males tweets from females tweets. Aside previous results which were not better compared to ours, it is well known in the real world that some males write or speak like their females counterpart and vice versa. We point to infer that

the algorithm in question has trouble distinguishing between female and male gender counts which is not much of a problem based on previous results which were not better compared to ours. All the true predictions are in the diagonal. It is therefore easy to visually inspect the prediction errors, as they are represented by values outside the diagonal.

4.3.1.1 MISCLASSIFICATION

The Kaggle dataset used for training is not from South Africa as compared to the tweets collected and currently streaming through Elasticsearch and `Kibana` web services. Hence, there is a variation in linguistic style which could have biased the classifer in both confusion matrix. The algorithm was exposed to more female samples than males, suggesting a bias during training. The misclassification count for males is higher than that for females supporting the earlier stated premise. For future work, we suggest using stratified sampling to ensure that gender is equally represented.

4.4 SUMMARY

Against stated literature in Chapter 2, the gender prediction model of this research largely depends on a framework which broadly simplifies machine learning (ML) algorithms, especially deep learning (DL) models from conceptualising, training, evaluating and export for serving into production. This framework in Chapter 3 is integrated within TensorFlow (TF) and it is called the Estimator. When building neural networks in TF, it is important to use a builder API because it facilitates the implementation and modification of the source code. This reduces the risk of bugs as well. The Estimator is that high-level API within TF that implements ML models for training a predictive outcome to be used on an unseen data. In this case, we reveal the gender from a given set of real-time streaming sexual violence unstructured tweets through our implemented DL approach on top of TF. The next chapter presents the overview of the Web application. It explains sentiment analysis through processing the linguistic style preferred by a Twitter user from a polarity-oriented approach.

CHAPTER 5

SYSTEM REVIEW AND DISCUSSION

5.1 INTRODUCTION

The preceding chapter evaluated the performance of the model through graphs and generation of confusion matrix on the test and training sets. On gender-based violence, this chapter is focused on incidents of sexual violence in South Africa. The system is currently hosted on a private cloud service - `digital ocean` and streams live tweets on sexual violence (regarding hidden attributes such as gender and sentiment analysis). This chapter attempts to present an overview of the system through the visualisation of real-time sexual violence tweets from the development of an intuitive dashboard on `Kibana` (the core visualizer of the system). It presents results and discusses real-time streaming tweets on sexual violence.

5.2 THE SYSTEM

We present a fully-fleshed web-based system displaying the application of AI and web services for streaming sexual violence tweets as a form of `GBV`. This enabled the discovery of which gender reported an instance of sexual violence and the sentiment in the form of generated opinions. We provided comprehensive information on the dual use of `ES` and `Kibana` in the literature in Chapter 2. `Kibana` as an open source data visualisation tool was utilised for visual exploration as well as real-time analysis of a user's data in `ES`. `ES` is a search engine that provides capabilities such as

queries and storages. It further enables users to create various outputs such as graphs and maps from large volumes of text data. In this section, Figure 5.1 is a visual representation of the `Kibana` menu bar which depicts within it the various toolbars/links that are available and will be expanded on in the sections below. Therefore, we present the three core components to which the visualisations in this research are embedded and easily accessed.

1. Discover
2. Visualize
3. Dashboard

5.2.1 DISCOVER

This tab enables visual exploration of one's tweet as shown in Figure 5.2. It includes details such as date and time of when a tweet relating to sexual violence was created, the user's description, actual tweet text and location of the tweet author, which is shown in Table 5.1 and Figure 5.3. Through this tool, a number of queries can be explored and the relevant information around that query will be presented e.g. the author's gender query shows that 55.8% of the tweets were from females, 32.2% from males and 12% from brands (non-individual). The Sentiment query shows the percentages of negative, positive and neutral tweets as shown in Figure 5.1. It is the cumulative sentiment variables which revealed various levels of characterised differences collected via tweets from females and males respectively as further shown in Figure 5.5 and 5.6. The percentage measures attributed to both gender suggested interesting statistics. For example, the variations from the angle of sentiment analysis showed that the female gender is more inclined to engage with cases that is related gender-based violence. This does not entirely discredit the male gender as indifferent in this regard, but the measure in sentiment favours the female gender from the view point of an active demographic entity. This is influenced by indicators where the female gender is understood to be more vulnerable, hence, exhibiting more sentiment in this matter.

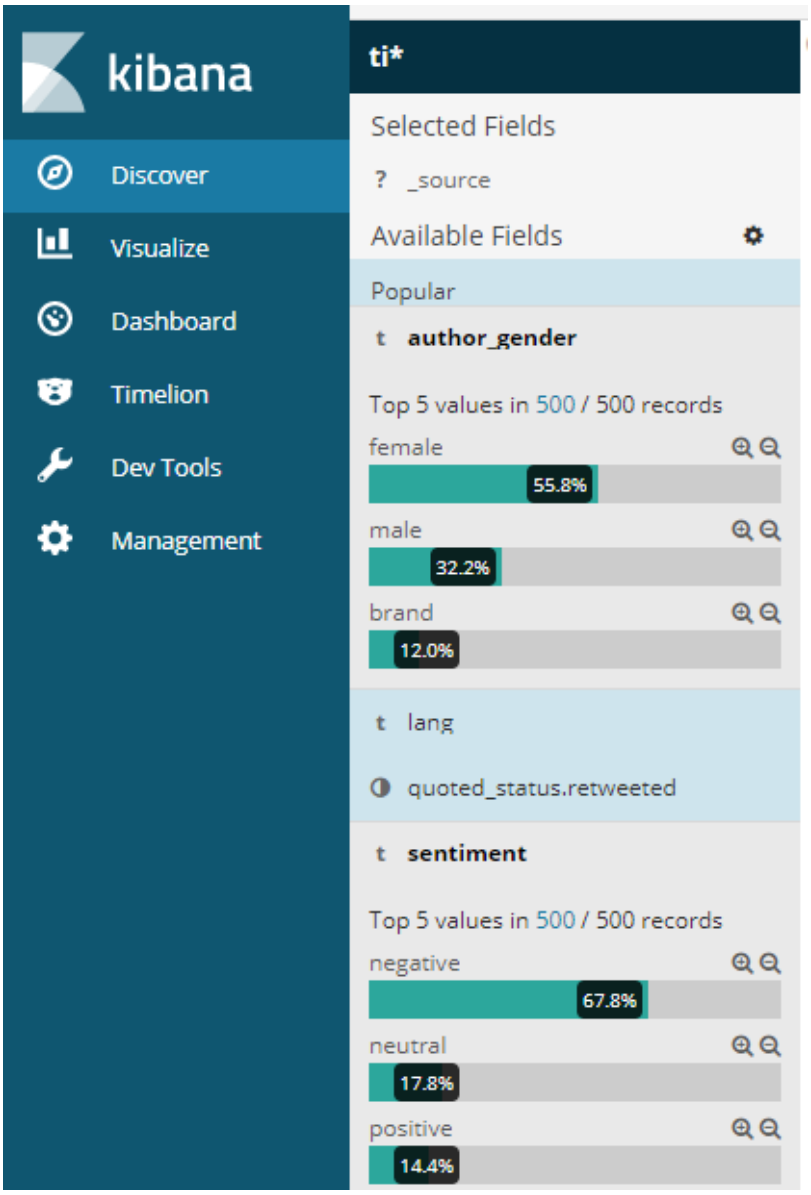


FIGURE 5.1: The Kibana's interface showing cumulative estimates of gender variables and sentiment scores



FIGURE 5.2: Real-time tweet sample from the Discover menu bar

TABLE 5.1: Columns on feature extractions from Kibana

user.description	sentiment	text	author_gender	user.location

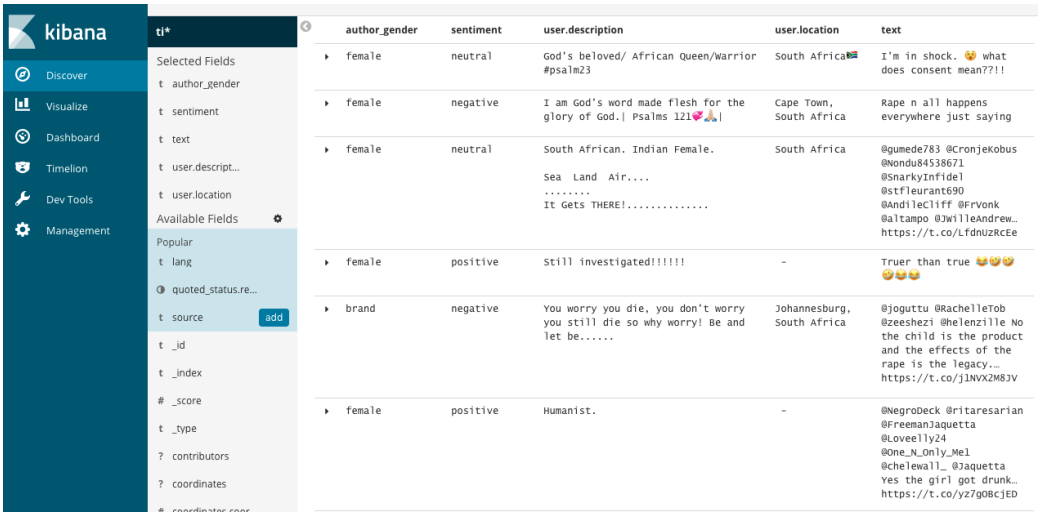


FIGURE 5.3: Feature extractions from the Kibana menu

5.2.2 VISUALIZE

This tab in Figure 5.1 has various graphical representations for data exploration and investigation. For instance, vertical bars are used to represent the cities where tweets originated from, region

map to show country of origin of the authors, pie chart to show gender contribution, tag cloud to show the most prominent activists with respect to sexual violence and a heat map to show the sentiments. Clicking on each graphical representation will bring up the actual graph and hovering over the image will produce respective figures relating to the data represented in the graph. See below Figure 5.5 and 5.6:

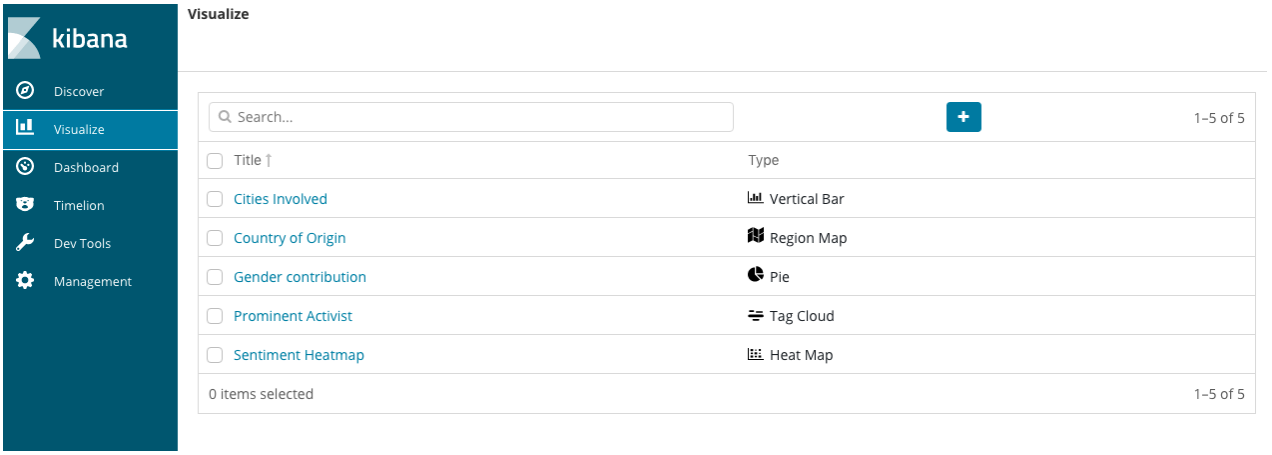


FIGURE 5.4: Graphical representation showing menus of visualizations

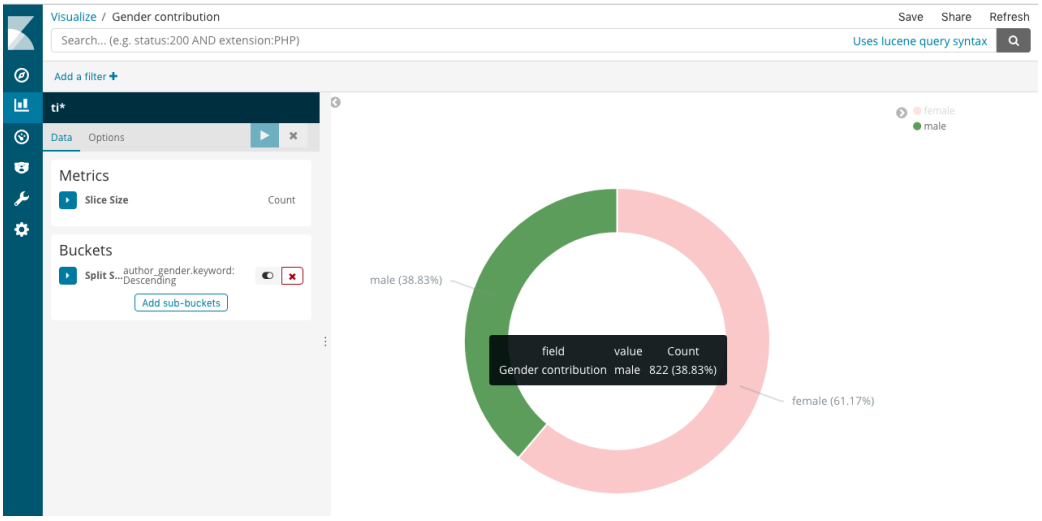


FIGURE 5.5: Male contribution on sexual violence tweets

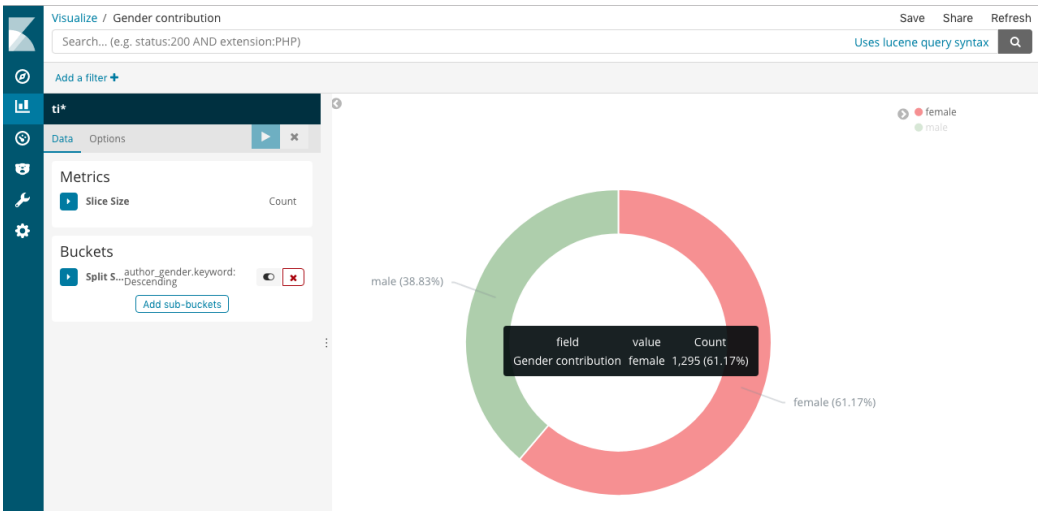


FIGURE 5.6: Female contribution on sexual violence tweets

5.2.3 DASHBOARD

This tab shows all graphical representations in one work space. Aesthetic changes can be made to individual graphs in this workspace. Overall, in Figures 5.7 and 5.8, this dashoard/work space is an integrated space allowing one to access data, explore it and visualise accordingly.

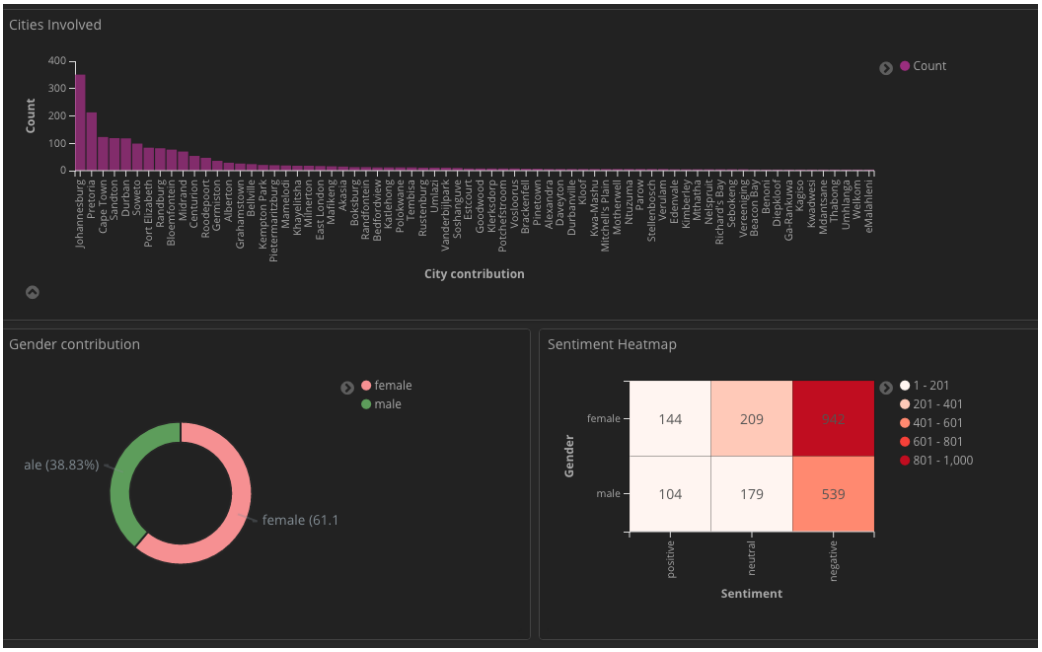


FIGURE 5.7: Dashboard 1: This reflected on the city with most reported cases in terms of gender and sentiment represented

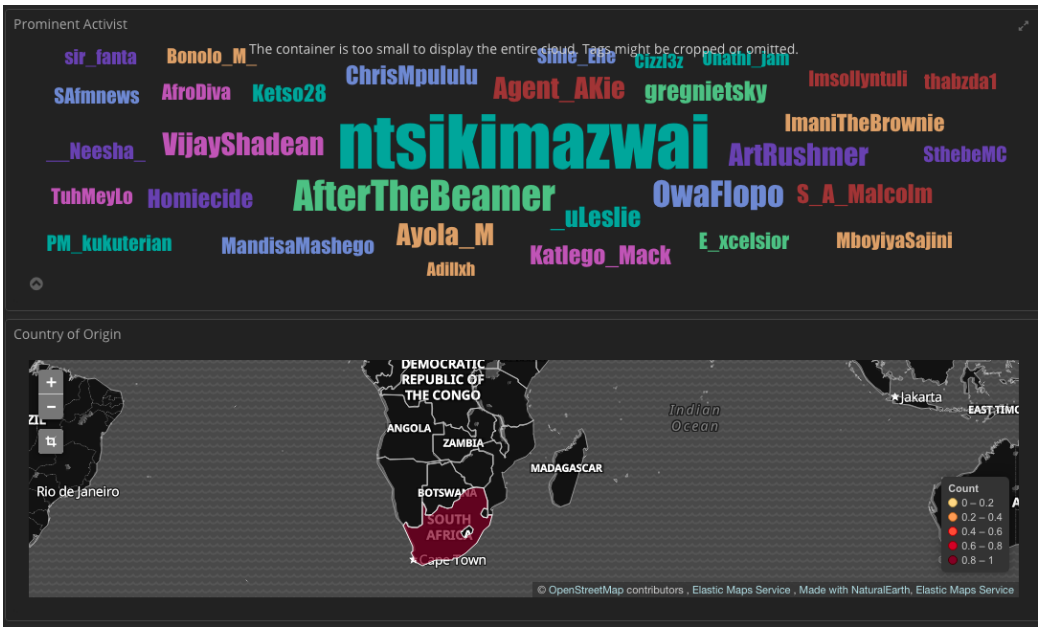


FIGURE 5.8: Dashboard 2: This presents the handles in South Africa with most frequency of tweets indicated in bold

5.3 DISCUSSION

5.3.1 TEXTUAL DISCUSSION

From a pool of unstructured information found on Twitter, sexual violence tweets were chosen as a representative subject. The research revealed that, by using an investigative approach and leveraging on deep neural network libraries coupled with the right processing of data, hidden attributes such as gender and sentiments could be inferred from any form of data. As expected, this information about sexual violence often spark insights about people's perceptions and the demographics affected. Consequently, the information about whether the people tweeting are victims or not became irrelevant in this instance. Thus, the data collected is still a valid social indicator because the effects of sexual violence cut across all societal boundaries. This is useful piece of information from a sociological point of view in that the report of these cases on Twitter is not only valid if it comes from a victim but that it can equally be valid if it is comes from a friend of a victim or family of a victim. Given this premise, we show:

1. How using a real-time web-based application was developed in mining sexual violence tweets. This is a powerful tool by the fact that we were able to extract information which expanded beyond merely reporting on an academic subject. This information is captured in Figures 5.1, 5.2 5.3, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10 and 5.11. They ultimately reiterated the significance of the application for social good.

2. The prospect of our AI-driven web-based application in mining, collecting and analysing meaningful information to be used for further analysis on sexual violence as a form of gender-based violence in South Africa.

5.3.2 VISUAL DISCUSSION

Figure 5.9 below is a bar graph showing a section of the multiple cities in South Africa where incidences of gender-based violence (GBV) were reported. Johannesburg and Pretoria were the two cities that exhibited the highest number of cases recorded with values of 351 and 213 respectively. Together with cities such as Durban and Cape Town, the number of cases brought to light made up 33.5% of the total number of cases reported. Areas such as Goodwood, Brackenfell and Alexandra, to name a few, had low reporting values as 8, 7 and 6 cases being reported.

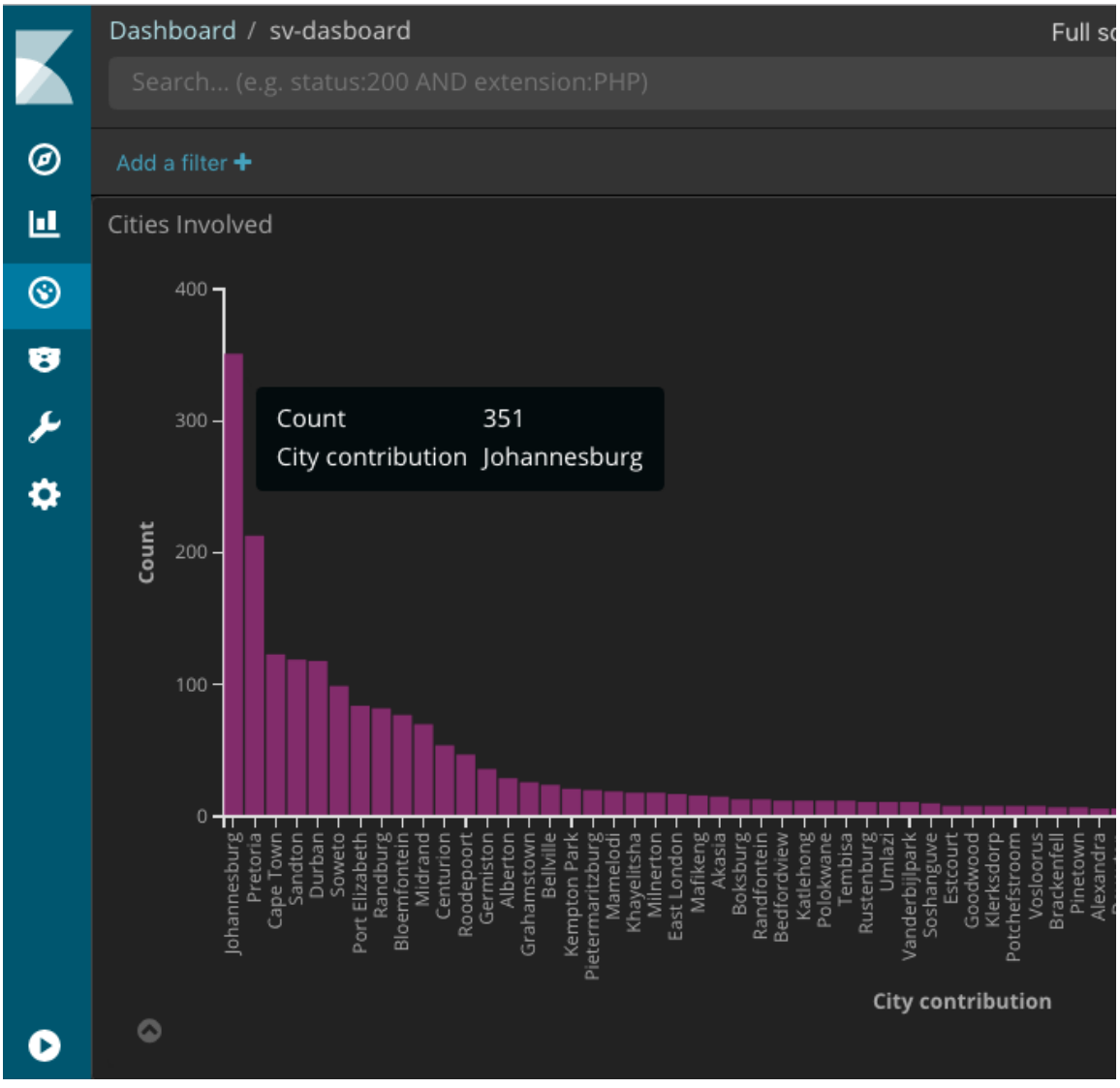


FIGURE 5.9: These are some of the cities showing origin of tweets

Figure 5.10 below is a pie chart showing the percentage of tweets on sexual violence contributed by each gender. 61.17% of the tweets were from females with 38.83% emanating from the male gender. These figures show a huge disparity between the genders tweeting about sexual violence. We addressed possible reasons for this in Section 1.3 of Chapter 1.

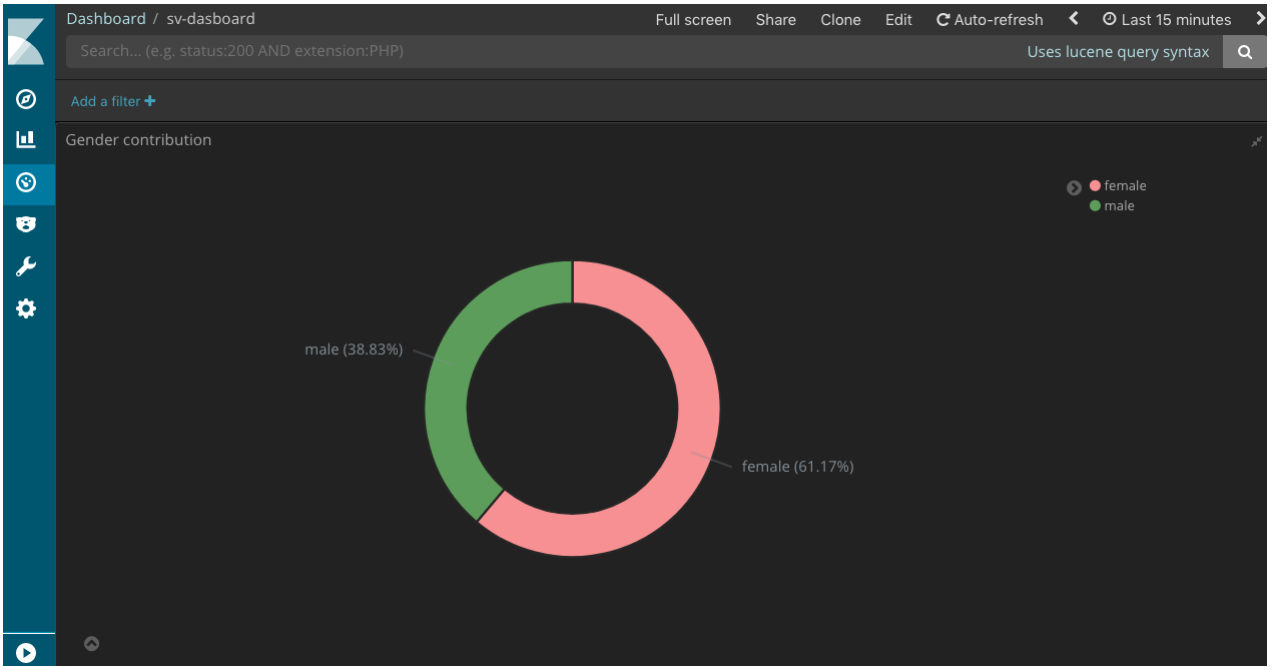


FIGURE 5.10: A Pie chart showing contributions from male and female

In Figure 5.11, we present a heatmap as a graphical representation of data represented by a map. Data is displayed in different shades of colours, the thickest colour representing the most prominent element and the lowest shade representing the least popular element. From our heatmap, we can see that the most prominent tweets were negative ones tweeted by females. Following this pattern, the least prominent tweets were those with a positive context which were tweeted by males. For an aggregated metric in percentage, refer to 5.10 which reflected the total metric size in percentage.

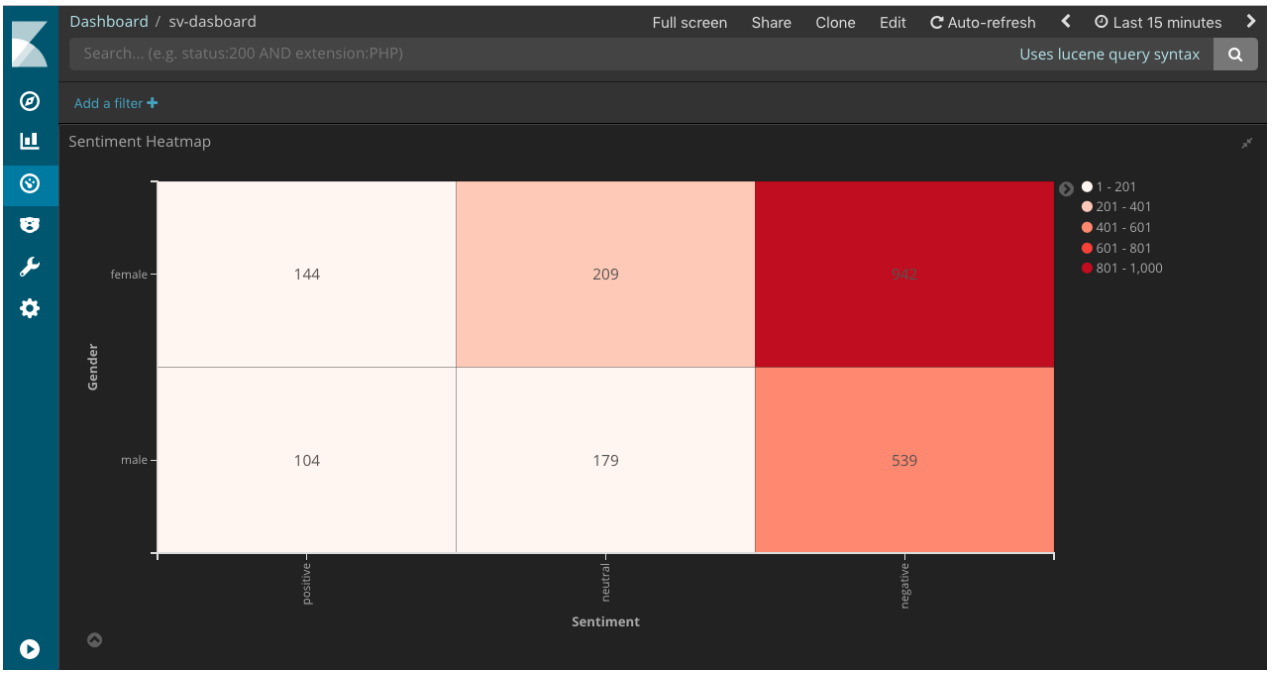


FIGURE 5.11: A Heat map showing sentiment count

Table 5.2 further suggested that a large proportion of female reported more on cases related to GBV by an estimate of 72.74% through tweets on Twitter in comparison to their male counterpart who only reported 65.57%.

TABLE 5.2: Sentiment count from both gender classes

Gender	Sentiment	Estimated value (%)	Total sample (count)
Female	Positive	11.12	1295
	Neutral	16.14	
	Negative	72.74	
Gender	**Sentiment**	**Estimated value (%)**	**Total sample (count)**
Male	Positive	12.65	822
	Neutral	21.78	
	Negative	65.57	

This is a fact-based statement in that females showed a negative sentiment of 72.74% versus males who showed a negative sentiment of 65.57%. This result reflects a good validation of our experiment as we would expect more negative comments from women than men that is disproportionately affected by sexual violence. The measures of disparity are not an artefact of the system, but that these numbers make sense in the real-world because women are more negatively impacted by sexual violence. So, it is expected to have more comments that reflects negative sentiments from women.

TABLE 5.3: A description of relative fields from tweet document

Fields	Description
user_description	This explains the biography embedded within a user's Twitter profile. This gives information which is inherent to a particular user and describes the attitude in a certain area of preference.
sentiment	This field reveals the actual opinion exclusive to a given user. With reference to the choice of language in its written form, the sentiment orientation of a tweet text was verified. This exposes the impression made on the subject by an author.
text	This field exposes the style of the author in language usage to be analysed for further discoveries.
author_gender	This field confirms the gender that has tweeted in an instance of sexual violence. This field actually concludes on this research as it reveals that given the bio (user profile description) and text property of a single user, we can train a deep neural network model to predict the gender of an author that has contributed to the subject.
user_location	This confirms South Africa as our primary geographical consideration of an author and when a tweet was posted within a specific city.

The `Kibana` navigation as referenced in Figure 5.1 and shown in Figure 5.3 showed key attributes extracted from streaming tweets on sexual violence as presented in Table 5.1. Aside from further discussions which will be centred on the rationale of gender prediction and sentiment analysis, we describe relative fields in Table 5.3.

5.4 RATIONALE OF GENDER PREDICTION AND SENTIMENT ANALYSIS

In this research, we were able to use machine learning (a subset of AI) and Web services to predict gender from real-time streaming of sexual violence tweets and verify sentiment. The first approach undertaken in this research was to ingest tweets on sexual violence through the connection of Twitter streaming API and the `Indexer`. Automating data collection and analysing its extractive information with visualisations to reveal insights that were previously unknown suggests that our lives and activities are data-driven.

To the best of our knowledge, we were able to show at scale how the research questions were answered and by extention, the problem description.

5.4.1 INFERRING GENDER

In this research, we have explored various techniques in an attempt to predict gender and analyse sentiments from GBV related tweets from the Twitter platform. Our approach is complex by the fact that we built an AI model through the deployment of a deep learning approach with Google's TensorFlow for data abstraction. The model was trained on labeled datasets to be used on a real-time unlabeled dataset. The AI model is able to discriminate between classes after training with labeled data. The label could be either male, female or brand (a non-individual). From a particular Twitter account, the profile description and text field of a tweet object were chosen because it exposes the writing style of the author. In the context of a social conversation, the aim of the AI model was to try to discriminate between the writing styles of both male and female and also to predict gender given an out-of-sample data set. This was achieved from a model trained on the text and bio fields (profile description). We stressed that the focus here is not on accuracy without disregarding its potential of affecting how the model performed in comparisons to similar models, but on how we implemented transfer learning through TensorFlow Hub layers in a model for text classification.

5.4.2 INFERRING SENTIMENT ANALYSIS

The aim of the sentiment analysis in this research project is to evaluate the attitude of a speaker to a particular issue or the overall contextual clarity of a text. We were able to answer this aspect of the research question from the view of approach and significance. By approach, we mean the procedure in developing a model to help us answer the question, and by significance, we mean, the relevance in developing a system to help us answer this question. Our application called sentiment analyser with its robust libraries embedded in JavaScript was used to enable communication on the `Node.js` runtime. The approach used is polarised-oriented from English words in the AFINN model dictionary. The idea around this is that given a tweet text, the application will sum up weights for both positive, negative and neutral words. The sentiment class with the highest value is summed up and this reflects the sentiment of the text. We undertook sentiment analysis of English language tweets to basically identify and extract actual cases of an instance of sexual violence.

5.5 SUMMARY

This research achieved an improved approach into mining Twitter data for social good through the development of a Web-based system for ingesting, querying, analysing and visualising sexual violence related tweets from Twitter. We showed in screenshots, and excerpt from the system that predicting gender and analysing sentiment in a tweet document delivered significant results whose relevance scaled beyond a scholarly work. The next chapter concludes this research.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

The Estimator as a high-level API within TensorFlow module exposes a wide variety of machine learning (ML) algorithms through the integration of state-of-the-art ML models for large scale supervised or unsupervised problems. We have used it as a task-oriented interface in simplifying a unique use case in ML for data abstractions on gender classification. The Estimator simplifies ML applications through the usage of general-purpose high-level language as building blocks for approaches specific to any level of use case. For example, in a situation like medical imaging [Michel et al., 2012]. Emphasis is on simplicity of implementation, performance, documentation, and API consistency, encouraging its use in an academic space.

When working with a highly fluid, fast-moving domain like Twitter - populated by users who may unite around a topic and engage in volatile communication, it is crucial for researchers to be thorough about the extent of insight to which extracting and analysing its data would reflect. As a cloud-based repository on text, Twitter generates, processes, and stores data in a measure that far exceeds its capacity to efficiently analyse and evaluate massive volumes of data in variations of topics and components. By reason of the fact that the measure of this data is largely increasing, it is impractical when it comes to considering the analysis of the data without using sophisticated designs and applications. Therefore, this valuable information mined from Twitter in its raw and unstructured state is fundamental in the evaluation measure of algorithms used. As such, the information mined will enable us to confront and resolve complex situations such as sexual violence,

which would in turn afford us an edge in tackling sensitive issues in other forms of gender-based violence.

Moreover, there are very few avenues for getting data collection about sexual violence; there is under reporting and most people are hesitant to share details of their ordeal with the authorities [Coelho, 2019]. However, people are more comfortable sharing on Twitter [Mendes et al., 2018]. In addition, this research has yielded interesting results, one of which is the disparity on negative sentiment expressed by the female gender class when reporting about tweets on sexual violence. This is a discovery that has scaled beyond the computer science academic frame but is a piece of information that would be of great use to a sociologist. Therefore, we are fundamentally producing a new database that will contribute to future studies that want to expand on this subject or related issues.

6.2 RECOMMENDATION FOR FURTHER STUDY

We have managed to achieve the prediction of gender as a demographic indicator given Twitter data, but the study as a whole is still on-going. Nonetheless, the work was able to produce demographic data around people tweeting about sexual violence in the hope that it will be able to validate real-world cases with the data generated from our system. With that said, it is clear that this research is a piece of a larger work that aims at understanding gender-based violence in general and provides a very significant insight about gender and sentiment. More importantly, we have built a framework regarding the collection and classification of Twitter data that can then be used to infer other demographic determinants, variables, and indicators. For instance, education level. As a result, if someone else does it, they do not have to start from the scratch as there is now a system that was produced from this research which can serve as a data source.

6.2.1 TECHNOLOGY

1. Further studies, using NLP methods such as topic modelling can analyse text content and provide a better understanding of public perception on this issue. At present, our study does not look at key topics discussed in the tweets and does not shed much light on why users expressed a particular sentiment.

2. Using batch processing parallel to stream processing to re-calibrate model parameters.

    - This research is not within the scope and version of Twitter's Firehose API. This is because of its expensive demand in setting-up and the inability of this research to scale beyond what is currently afforded. Twitter's restrictions of downloadable tweets is using

Twitter's streaming API which allowed 1% of tweets to over 40% of tweets on a real-time basis.

3. Compare alternative methods to feature selection from text TF-IDF etc.

4. As a limitation, we could not further improve on the accuracy of the training set in a bid to generalise on the model owing to a constraint on acquiring more Graphical Processing Units (GPUs) and Tensor Processing Units (TPU) which was not cheap at the time of this research.

   Cross validation is computationally expensive for a neural network model.

   The experiment carried out on sentiment analysis using the proposed model did not quite negate tweet text when fed into the sentiment analyser.

5. A system like ours with the rationale for the development of models should be protected within a secured socket layer (SSL). This system is built to assist stakeholders such as law enforcement agencies, the police, and human rights institutions with a unified communicating and tracking medium that addresses societal threatened problems and it requires cloud-based authentication for its level of robustness. We recommend the cluster which currently holds our running web application be made to run on a secured web socket for privacy control.

6. If a more sophisticated tracking system of some sort is integrated into the model such that it expands on the possibility of closely monitoring reported patterns, and perpetrated acts of GBV in South Africa, it will be possible to assist victims; or at least come close to being able to protect the vulnerable and abused persons. However, research collaborations with the school of humanities should be encouraged to continuously expand the knowledge that will lead to the development of an improved system that can better assist in managing and solving GBV related crimes.

6.3 CONTRIBUTIONS

The research revealed that, by using an investigative approach and leveraging on deep neural network libraries coupled with the right processing of data, hidden attributes such as gender and sentiments expressed as opinions in text documents could be inferred from any form of data. Our contributions through experiments are further explained below.

1. South Africa is increasingly realising the significance and effect of GBV and the need to strengthen the sector-wide response. Through preventive initiatives by way of revealing hidden attributes like gender from a tweet document on gender-based cases, law enforcement agencies can take advantage of the statistics and disparity as a multi-faceted response on a complex issue as GBV.

2. Through experiments performed, results showed cities in South Africa where this issue of reported cases is most prominent. This will point interested parties who are concerned with eliminating cases of GBV in South Africa to wisely allocate resources to curb the menace.

3. Location could be said to have been embedded in tweets, however, our system further emphasised its significance for a reported GBV instance.

4. The date and time of the reported occurrance of sexual violence related cases was captured. This made our system robust by the fact that we can actually point to a moment of report and perpetration relative to a tweet.

5. You will be able to deduce the sentiment analysis by way of evaluating opinions given each tweet document.

Therefore, these findings have the potential to help policy makers and parties with interest in GBV in South Africa to initiate focused-decision measures to drive change in this regard. Through these, we conclude that this research has made significant contributions to science and social science alike.

APPENDIX A

PRE-TRAINING THE GENDER CLASSIFIER

LIBRARY DEPENDENCIES

```
1
2 import tensorflow as tf
3 import pandas as pd
4 from nltk.corpus import stopwords
5 import numpy as np
6 import tensorflow_hub as hub
7 import os
8 import ref
```

LISTING A.1: Library Dependencies

PRE-PROCESSING

The code below is used in pre-processing. Essentially, *stop words* and *punctuation marks* including *hashtags* and *url links*. The text case is then standardised.

```
9  #This functions removes the common english stopwords like and and the, since
       they do not really
10 def remove_stopwords(df, field):
11 stop = stopwords.words('english')
12 df[field].apply(lambda x: [item for item in x if item not in stop])
```

```
13 return df
14
15 #This function cleans the text, by removing symbols and spaces
16 def normalize_text(s):
17 # just in case
18 s = str(s)
19 s = s.lower()
20
21 # remove punctuation that is not word-internal (e.g., hyphens, apostrophes)
22 s = re.sub('\s\W',' ',s)
23 s = re.sub('\W\s',' ',s)
24
25 # make sure we didn't introduce any double spaces
26 s = re.sub('\s+',' ',s)
27
28 s = re.sub(r'[^\w]', ' ', s)
29 s = re.sub("\d+", "", s)
30 s = re.sub('[!@#$_]', '', s)
31 s = s.replace("co","")
32 s = s.replace("https","")
33 s = s.replace(",","")
34 s = s.replace("[\w*"," ")
35
36 return s
```

LISTING A.2: Data Pre-Processing

GETTING TRAINING DATA

As mentioned, the gender predictor is trained on a Kaggle dataset on gender classification. The code below ingests the dataset as a **Pandas** DataFrame, runs pre-processing and encodes the gender into numerical values. It splits the tweets into two sets i.e, the training set - *used to generalise characteristics/features that determine gender from tweets* and the testing set - *used to evaluate how well the algorithm can predict on unseen data.*

```
38
39 def load_data_from_file():
40 df = pd.read_csv("./data.csv", encoding='latin1')
41 df = df.loc[df['gender'].isin(['male','female', 'brand'])]
42 df['gender'] = [normalize_text(s) for s in df['gender']]
43 df['gender'] = df['gender'].map({'male':0, 'female':1, 'brand':2})
44 df['gender'] = df['gender'].astype('int')
45 df['text_norm'] = [normalize_text(s) for s in df['text']]
46 df['description_norm'] = [normalize_text(s) for s in df['description']]
```

```
47 df['all_features'] = df['text_norm'].str.cat(df['description_norm'], sep=' ')
48 #df['all_features'] = df['description_norm']
49 #df = remove_stopwords(df, 'all_features')
50
51 train, test = np.split(df.sample(frac=1), [int(.8 * len(df))])
52 train = train.loc[:,['all_features','gender']]
53 #validate = validate.loc[:,['all_features','gender']]
54 test = test.loc[:,['all_features','gender']]
55
56 # Splits the data into training and testing datasets
57 print("train size", len(train))
58 print("test size", len(test))
59 #print("validate size", len(validate))
60 return train, test
61
62 # Use load_data_from_file function to get pre-processed data
63 train_df, test_df = load_data_from_file()
```

LISTING A.3: Getting Data

HELPER FUNCTIONS

The functions below are used by the training and testing method which are discussed next. In order in which they appear, they provide the following functionalities:

Providing a container for the trainer dataset.

Providing a container for the trainer testing.

Predicting gender using a single input from the training data.

Predicting gender using a single input from the testing dataset.

Feeding/Serving the algorithm data in batches for training.

```
65
66 # Build function to hold training input data
67 train_input_fn = tf.estimator.inputs.pandas_input_fn(train_df, train_df["gender
       "], queue_capacity=500, num_epochs=None, shuffle=True, batch_size=64)
68
69
70 # Building function to hold testing input data
71 predict_test_input_fn = tf.estimator.inputs.pandas_input_fn(test_df, test_df["
       gender"],  queue_capacity=500, shuffle=False,  batch_size=64)
72
```

```
# Builds a function for testing a single example
def predict_single_input_fn():
d = pd.DataFrame.from_dict(data = {"all_features": ["It gets better with every angle"], "gender":[0]})
predict_single_input = tf.estimator.inputs.pandas_input_fn(d, d['gender'], shuffle=False,  batch_size=64)
return predict_single_input


# Defines a function to feed data in batches to the algorithm
def serving_input_receiver_fn():
"""An input receiver that expects a serialized tf.Example."""
input_placeholder = tf.placeholder(dtype=tf.string,
shape=[None],
name='all_features')

input = {'all_features': input_placeholder}
return tf.estimator.export.ServingInputReceiver(input, input)

# Transform the all_features field into an embedding vector
embedded_text_feature_column = hub.text_embedding_column(key="all_features",
module_spec="https://tfhub.dev/google/nnlm-en-dim128-with-normalization/1",
trainable = False)
```

LISTING A.4: Helper Functions

TRAINING THE ALGORITHM

```

# Builds the estimator which represent the neural network
estimator = tf.estimator.DNNClassifier(hidden_units=[500, 150],
feature_columns=[embedded_text_feature_column],
n_classes=3,
model_dir="./model",
optimizer=tf.train.ProximalAdagradOptimizer(learning_rate=0.06, l1_regularization_strength=0.004))


# invoke the nueral network passing in the input function built earlier
# Training for 1,000 steps means 128,000 training examples with the d efault batch size.
# This is roughly equivalent to 5 epochs since the training dataset contains 25,000 examples.
estimator.train(input_fn=train_input_fn, steps=5500)

```

```
### NB: Determine Training accuracy
- Consider using sampling with replacement depending on the size of your data
```

LISTING A.5: Training Phase

TESTING THE ALGORITHM

```

#  Evalute neural network by input the testing input data
test_eval_result = estimator.evaluate(input_fn=predict_test_input_fn)

# Use the neural network to predict a single input
predict_tensor = estimator.predict(input_fn=predict_single_input_fn())

for result in predict_tensor:
print('result: {}'.format(result))

# print evaluation results
print("Test set accuracy: {accuracy}".format(**test_eval_result))
```

LISTING A.6: Testing Phase

SAVING MODEL AND LEARNED PARAMETERS FOR FUTURE USE

```
# save model
estimator.export_savedmodel("./model", serving_input_receiver_fn)
```

LISTING A.7: Testing Phase

APPENDIX B

HELPER FUNCTIONS

IMPORTING DEPENDENCIES

The code below requires the following dependencies for Node.js development:

Elasticsearch The role of `ES` in our project is to index and help in search and analysis of tweet data (sexual violence related tweets).

gRPC gRPC server uses Protocol Buffers (Protobuf) as its communication interchange format by default configuration. gRPC is an open, high-performance remote calling process (RPC) framework running on HTTP/2 from Google.

Lodash By taking the trouble out of working with arrays, numbers, objects, strings etc., Lodash makes JavaScript easier for this research.

Sentiment The sentiment is a Node.js module using the wordlist AFINN-165 and Emoji Sentiment Ranking to perform sentiment analysis on arbitrary input text blocks.

```
127
128 _ = require('lodash')
129 let elasticsearch = require('elasticsearch');
130 var Sentiment = require('sentiment');
131
132 # Instantiate the Sentiment Module
```

```
133 var sentiment = new Sentiment();
134
135
136 // this module is dicussed seperately in Appendix C
137 const gd = require("./lib/predict_module")
```

LISTING B.1: Library Dependencies

CONFIGURING THE ELASTICSEARCH API

The code below creates an object that will authenticate against the Elasticsearch API in-order to write data.

```
138
139 // Connect to elasticsearch client SDK
140 let client = new elasticsearch.Client({
141 //host: 'http://165.227.37.131:7501/app/kibana',
142 host: 'http://165.227.37.131:7501/app/kibana',
143 //log: 'trace'
144 });
145
146
147 // Create elasticsearch index if it doesn't exist
148
149 client.indices.exists({
150 index: 'ti'
151 }).then(exists => {
152 if (!exists) {
153 client.indices.create({
154 index: 'ti'
155 }, function (err, resp, status) {
156 if (!err) {
157 console.log("create", resp);
158 }
159 });
160 }
161 }).catch(error => {
162 console.log(error)
163 })
```

LISTING B.2: Library Dependencies

INSTANTIATE GOOGLE REMOTE PROCEDURE CALL (GRPC)

Serving is a concept you become familiar with in the event you want to deploy a trained and serve machine learning model into production. The validated model from training and prediction is served into a production environment by delivering the models to the TensorFlow Serving instances from the model's repository.

```
165
166 // Instantiate Google Remote Procedure Call (gRPC) client for TensorFlow
        Serving server.
167
168 const PROTO_PATH = __dirname + '/protos/prediction_service.proto';
169 // loading service proto
170 var tensorflow_serving = grpc.load(PROTO_PATH).tensorflow.serving;
171
172
173 // creating gRPC service client
174 var client = new tensorflow_serving.PredictionService(
175 //'http://165.227.37.131:7501/app/kibana',
176 'http://165.227.37.131:7501/app/kibana',
177 grpc.credentials.createInsecure()
178 );
```

LISTING B.3: Library Dependencies

TRANSLATING SCORES TO TEXT

```
179
180 // Utility function to translate sentiment score to class
181 const interpreteSentiment = function (score) {
182 if (score < 0)
183 return 'negative';
184 if (score > 0)
185 return 'positive'
186 if (score == 0)
187 return 'neutral'
188 }
189
190 // Utility function to translate gender values to class
191 const interpreteGenderClassification = function (score) {
192 if (score == 0)
193 return 'male';
194 if (score == 1)
195 return 'female'
```

```
if (score == 2)
return 'brand'
}

\end{itemize}
```

LISTING B.4: Library Dependencies

APPENDIX C

STREAMING SENTIMENT PREDICTION

IMPORTING DEPENDENCIES

Ingesting API calls demands the installation of "twit" library before the initiation of the require() package. The "twit" library is a Twitter API client for Node.js and it facilitates supports for both the REST and Streaming API.

```
201 const Twit = require('twit')
```

LISTING C.1: Library Dependencies

CONFIGURING THE TWITTER API

The code below create an object that will authenticate against the Twitter API in-order to extract data.

```
202
203 tweet_count = 0;
204 console.log(gd)
205 // Setup Twitter API Credentials
206 const T = new Twit({
207 consumer_key: 'XXXXXXXXXX',
208 consumer_secret: 'XXXXXXXXXX',
```

```
access_token: 'XXXXXXXXX',
access_token_secret: 'XXXXXXXXX',
timeout_ms: 60 * 1000, // optional HTTP request timeout to apply to all
    requests.
})
```

LISTING C.2: Library Dependencies

TWEET VALIDATION UTILITIES

The code below provides a utility functions that:

Validate whether a result return by the Twitter API is a valid tweet.

Ensures that the collected tweets are from South Africa through using the geolocation.

Detects tweets relating to GBV through using pre-defined keywords.

```

// Utility function to test if a tweet is valid
const isTweet = _.conforms({
user: _.isObject,
id_str: _.isString,
text: _.isString,
})


// Setup tweet Area of Interest bounding box, in this case south_africa
const south_africa = ['16.3449768409', '-34.8191663551', '32.830120477', '
    -22.0913127581']

// Setup filter to track sexual violence tweet
var stream = T.stream('statuses/filter', {
track: 'rape', 'sexual abuse', 'sexual assault', 'sexual harassment', 'sexual
    violence',
language: 'en'
})

\end{itemize}
```

LISTING C.3: Library Dependencies

STREAMING TWITTER DATA, PERFORMING SENTIMENT ANALYSIS AND GENDER PREDICTION

The code below builds a buffer that will continously pull data from the Twitter API

```
232
233 // Listen for tweets, on arrival annotate with sentiment and gender then index
        it
234 stream.on('tweet', function (tweet) {
235 if (tweet.place != null && tweet.place.country_code == 'ZA') {
236 gd(tweet.text + " " + tweet.user.description).then((result) => {
237
238 r = _.findIndex(result, i => {
239 return i == _.max(result)
240 })
241 tweet.sentiment = interpreteSentiment(sentiment.analyze(tweet.text).score);
242 tweet.author_gender = interpreteGenderClassification(r)
243 tweet_count++;
244 client.index({
245 index: 'ti',
246 type: 'tweet_data',
247 body: tweet
248 });
249 console.log(`${tweet_count} tweet indexed`)
250 })
251 }
252 })
```

LISTING C.4: Library Dependencies

APPENDIX D

DETAILS - INTEGRATING NODE.JS WITH PYTHON THROUGH TF-SERVING SERVER

gRPC AND TF-SERVING SERVER INTEGRATION

```
254 /**
255 * Calls predict gRPC method on TensorFlow Serving server by opening a buffer
        that
256 * continuously feeds the Tenserflow DNNClassifier with tweets as discussed in
        Appendix A.
257 */
258
259 predict = (buffer, fn) => {
260
261 // The buffer is an array of word embeddings (refer to Appendix A.4)
262 var buffers;
263 if (buffer.constructor === Array) {
264 buffers = buffer;
265 } else {
266 buffers = [buffer];
267 }
```

LISTING D.1: Library Dependencies

```
1
2 // building PredictRequest proto message
```

```
const msg = {
  model_spec: {
    name: 'tn'
  },
  inputs: {
    inputs: {
      dtype: 'DT_STRING',
      tensor_shape: {
        dim: {
          size: buffers.length
        }
      },
      string_val: buffers
    }
  }
};
```

PREDICT STREAMING INPUT USING THE DNNCLASSIFIER

```


// calling gRPC method to predict using the tenserflow DNNClassifier (Appendix A.6)

client.predict(msg, (err, response) => {
if (err) return fn(err);

//console.log(response)
// decoding response

const classes = response.outputs.classes.string_val.map((b) => b.toString('utf8'));
const scores = response.outputs.scores.float_val;


// Creating a list of predicted genders/results
var i,
len = classes.length,
chunk = 5,
results = [];
for (i = 0; i < len; i += chunk) {
results.push(classes.slice(i, i + chunk));
}

fn(null, scores)
```

```
});

}
```

LISTING D.2: Library Dependencies

EXPOSE API TO ALWAYS PROVIDE GENDER PREDICTION FROM TF-SERVING SERVER

```
// Return a function that promises to give gender prediction results given a
    tweet

module.exports = determineGender = (text) => {
var buf = Buffer.from(text, 'utf8');
return new Promise((resolve, reject) => {
predict(buf, (err, res) => {
if (err) {
reject(err);
}
resolve(res)
});
})
}
```

LISTING D.3: Library Dependencies

APPENDIX E

EVALUATING TRAIN-TEST-SPLIT-SETS ON THE GROUND TRUTH DATA

CONFUSION MATRIX FOR MODEL EVALUATION

ENVIRONMENT SETUP AND LIBRARY DEPENDENCIES

```
310
311 import tensorflow as tf
312 import pandas as pd
313 import matplotlib.pyplot as plt
314 from sklearn.metrics import confusion_matrix
315 from sklearn.utils.multiclass import unique_labels
316 from nltk.corpus import stopwords
317 import numpy as np
318 import tensorflow_hub as hub
319 import os
320 import re
321 import seaborn as sn
```

LISTING E.1: Environment Setup and Library Dependencies

PRE-PROCESSING FUNCTION

These functions essentially cleans and removes unwanted words like an article and URL(s) from text.

```
322
323 def remove_stopwords(df, field):
324 stop = stopwords.words('english')
325 df[field].apply(lambda x: [item for item in x if item not in stop])
326 return df
327
328 #This function cleans the text, by removing symbols and spaces
329 def normalize_text(s):
330 # just in case
331 s = str(s)
332 s = s.lower()
333
334 # remove punctuation that is not word-internal (e.g., hyphens, apostrophes)
335 s = re.sub('\s\W',' ',s)
336 s = re.sub('\W\s',' ',s)
337
338 # make sure we didn't introduce any double spaces
339 s = re.sub('\s+',' ',s)
340
341 s = re.sub(r'[^\w]', ' ', s)
342 s = re.sub("\d+", "", s)
343 s = re.sub('[!@#$_]', '', s)
344 s = s.replace("co","")
345 s = s.replace("https","")
346 s = s.replace(",","")
347 s = s.replace("[\w*"," ")
348
349 return s
```

LISTING E.2: Pre-Processing Function

EXPLORATORY DATA ANALYSIS

```
350
351 # Read in dataframe
352 df = pd.read_csv("./data.csv", encoding='latin1')
353 df.head()
354 # Inspect data
355 df.info()
```

LISTING E.3: Exploratory Data Analysis

LOOKING AT THE GENDER VARIABLE

We are only looking for classification between males and females, and the brand is not relevant in this case. So, we will filter the dataset of gender classes.

```

df.gender.value_counts()
# Find all tweets where the gender is either male or female
gender_filter = (df.gender == "male")| (df.gender == "female")
# filter the dataframe object
df = df[gender_filter]
df.gender.value_counts()
```

LISTING E.4: Looking at the Gender Variable

READING DATA FUNCTION AND LOOKING AT THE TEXT VARIABLE(S)

```

df.text.head()
df.description.head()


def load_data_from_file():

#Read file and filter by Gender in {"Male", "Female"}
df = pd.read_csv("./data.csv", encoding='latin1')
df = df.loc[df['gender'].isin(['male','female'])]

#Normalize Gender Field
df['gender'] = [normalize_text(s) for s in df['gender']]

#Binary Encode The Gender Field to get the target
df['gender'] = df['gender'].map({'male':0, 'female':1}).astype('int')


#Pre-Processing for Predictors i.e, text and the description

df['text_norm'] = [normalize_text(s) for s in df['text']]
df['description_norm'] = [normalize_text(s) for s in df['description']]
```

```
385
386 #Combine the text and description to one field
387 df['text'] = df['text_norm'].str.cat(df['description_norm'], sep=' ')
388
389 return df[['gender', 'text']]
390
391 #Using the function
392 dataset = load_data_from_file()
393 dataset.head()
```

LISTING E.5: Looking at the Gender Variable

PARTITION DATAFRAME INTO TRAINING AND TESTING

```
394
395
396 def train_test_split(data, train_percent = 0.8 ):
397 # Randomly shuffle the dataset and split into a train and test dataset
398 train, test = np.split(data.sample(frac=1), [int(train_percent* len(df))])
399 return(train, test)
400
401 train_df , test_df = train_test_split(data = dataset,
402 train_percent = 0.8
403 )
404
405 train_df.gender.value_counts()
```

LISTING E.6: Partition dataframe into Training and Testing

FUNCTIONS TO FEED DATA INTO THE MODEL

```
406
407
408 def data_feeder_fn(train_df, test_df):
409 # input function that will feed training DataFrame into the model
410 train_input_fn = tf.estimator.inputs.pandas_input_fn(
411 x = train_df,
412 y = train_df["gender"],
413 queue_capacity = 500,
414 num_epochs = None,
415 shuffle=True,
416 batch_size=64
417 )
```

```
# input function that will feed testing DataFrame into the model
test_input_fn = tf.estimator.inputs.pandas_input_fn(
x = test_df,
y = test_df["gender"],
queue_capacity = 500,
shuffle = False,
batch_size = 64
)

return(train_input_fn, test_input_fn)

train_input_feed, test_input_feed = data_feeder_fn(train_df, test_df)
```

LISTING E.7: Functions to feed data into the model

FEATURE EXTRACTION FROM THE TEXT FIELD

```


train_df.head()

# Transform the all_features field into an embedding vector
feature_columns = hub.text_embedding_column(
key="text",
module_spec="https://tfhub.dev/google/nnlm-en-dim128-with-normalization/1",
trainable = False)
```

LISTING E.8: Feature Extraction from the Text Field

CONSTRUCT DNNCLASSIFIER

```


tf.logging.set_verbosity(tf.logging.ERROR)
# Estimator representing neural network
estimator = tf.estimator.DNNClassifier(
hidden_units=[500, 150],
feature_columns=[feature_columns],
n_classes=2,
model_dir="./model_output",
optimizer= tf.train.ProximalAdagradOptimizer( learning_rate=0.06,
    l1_regularization_strength=0.004)
)
```

LISTING E.9: Construct DNNClassifier

TRAINING DNNCLASSIFIER

```
451
452 estimator.train(input_fn=train_input_feed, steps=5500)
453 metrics = estimator.evaluate(input_fn=train_input_feed, steps=100)
454 train_metrics = pd.DataFrame.from_dict(metrics,orient='index').T
455 train_metrics = train_metrics[["accuracy", "auc","precision","recall","loss"]]
456
457
458 train_metrics
```

LISTING E.10: Training DNNClassifier

TESTING DNNCLASSIFIER

```
460
461 test_metrics = estimator.evaluate(input_fn=test_input_feed)
462 test_metrics = pd.DataFrame.from_dict(test_metrics,orient='index').T
463 test_metrics = test_metrics[["accuracy", "auc","precision","recall","loss"]]
464
465
466
467 test_metrics
```

LISTING E.11: Testing DNNClassifier

PLOTTING THE CONFUSION MATRIX

```
469
470 # Get Predicted Values
471 test_prediction = estimator.predict(input_fn=test_input_feed)
472 test_prediction = list(test_prediction)
473 test_prediction =[ prediction["class_ids"] for prediction in test_prediction]
474 test_prediction = list(np.concatenate( test_prediction, axis=0 ))
475 test_prediction = np.array([ "Male" if x==0 else "Female" for x in
        test_prediction])
476
```

```
# Get Actual Values
test_actual = list(test_df.gender)
test_actual = np.array([ "Male" if x==0 else "Female" for x in test_actual])

def plot_confusion_matrix(cm,
target_names,
title='Confusion matrix',
cmap=None,
normalize=True):
"""
given a sklearn confusion matrix (cm), make a nice plot

Arguments
---------
cm:           confusion matrix from sklearn.metrics.confusion_matrix

target_names: given classification classes such as [0, 1, 2]
the class names, for example: ['high', 'medium', 'low']

title:        the text to display at the top of the matrix

cmap:         the gradient of the values displayed from matplotlib.pyplot.cm
see http://matplotlib.org/examples/color/colormaps_reference.html
plt.get_cmap('jet') or plt.cm.Blues

normalize:    If False, plot the raw numbers
If True, plot the proportions

Usage
-----
plot_confusion_matrix(cm           = cm,                  # confusion matrix
    created by
# sklearn.metrics.confusion_matrix
normalize    = True,                # show proportions
target_names = y_labels_vals,       # list of names of the classes
title        = best_estimator_name) # title of graph

Citiation
---------
http://scikit-learn.org/stable/auto_examples/model_selection/
    plot_confusion_matrix.html

"""
import matplotlib.pyplot as plt
import numpy as np
import itertools

```

```
accuracy = np.trace(cm) / float(np.sum(cm))
misclass = 1 - accuracy

if cmap is None:
cmap = plt.get_cmap('Blues')

plt.figure(figsize=(5, 5))
plt.imshow(cm, interpolation='nearest', cmap=cmap)
plt.title(title)
plt.colorbar()

if target_names is not None:
tick_marks = np.arange(len(target_names))
plt.xticks(tick_marks, target_names, rotation=45)
plt.yticks(tick_marks, target_names)

if normalize:
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]


thresh = cm.max() / 1.5 if normalize else cm.max() / 2
for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
if normalize:
plt.text(j, i, "{:0.4f}".format(cm[i, j]),
horizontalalignment="center",
color="white" if cm[i, j] > thresh else "black")
else:
plt.text(j, i, "{:,}".format(cm[i, j]),
horizontalalignment="center",
color="white" if cm[i, j] > thresh else "black")


plt.tight_layout()
plt.ylabel('True Gender')
plt.xlabel('Predicted Gender \n \n accuracy={:0.4f}; misclass={:0.4f}'.format(
    accuracy, misclass))
plt.show()


cm = confusion_matrix(test_actual, test_prediction)

plot_confusion_matrix(cm,
normalize    = False,
target_names = ['Male', 'Female'],
title = "Confusion Matrix for Gender Classification"
)
#sn.heatmap(cm, annot=True)
```

```
568
569
570 test_df.gender.value_counts()
```

LISTING E.12: Plotting of the Confusion Matrix

Bibliography

[Abadi et al., 2016] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[Abbasi et al., 2008] Ahmed Abbasi, Hsinchun Chen, and Arab Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):12, 2008.

[Abbasi et al., 2014] Ahmed Abbasi, Ammar Hassan, and Milan Dhar. Benchmarking twitter sentiment analysis tools. In *LREC*, volume 14, pages 26–31, 2014.

[AlSukhni and Alequr, 2016] Emad AlSukhni and Qasem Alequr. Investigating the use of machine learning algorithms in detecting gender of the arabic tweet author. *International Journal of Advanced Computer Science and Applications*, 7(7):319–328, 2016.

[Backstrom et al., 2010] Lars Backstrom, Eric Sun, and Cameron Marlow. Find me if you can: improving geographical prediction with social and spatial proximity. In *Proceedings of the 19th international conference on World wide web*, pages 61–70. ACM, 2010.

[Badjatiya et al., 2017] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 759–760. International World Wide Web Conferences Steering Committee, 2017.

[Bakliwal et al., 2012] Akshat Bakliwal, Piyush Arora, Senthil Madhappan, Nikhil Kapre, Mukesh Singh, and Vasudeva Varma. Mining sentiments from tweets. In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, pages 11–18, 2012.

[Barberá and Rivero, 2015] Pablo Barberá and Gonzalo Rivero. Understanding the political representativeness of twitter users. *Social Science Computer Review*, 33(6):712–729, 2015.

[Bartlett et al., 2002] Peter L Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48(1-3):85–113, 2002.

[Basow et al., 2007] Susan A Basow, Kristen F Cahill, Julie E Phelan, Kathryn Longshore, and Ann McGillicuddy-DeLisi. Perceptions of relational and physical aggression among college students: Effects of gender of perpetrator, target, and perceiver. *Psychology of Women Quarterly*, 31(1): 85–95, 2007.

[Benhardus and Kalita, 2013] James Benhardus and Jugal Kalita. Streaming trend detection in twitter. *International Journal of Web Based Communities*, 9(1):122–139, 2013.

[Berthon et al., 2012] Pierre R Berthon, Leyland F Pitt, Kirk Plangger, and Daniel Shapiro. Marketing meets web 2.0, social media, and creative consumers: Implications for international marketing strategy. *Business horizons*, 55(3):261–271, 2012.

[Buitinck et al., 2013] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. Api design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.

[Burger et al., 2011] John D Burger, John Henderson, George Kim, and Guido Zarrella. Discriminating gender on twitter. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1301–1309. Association for Computational Linguistics, 2011.

[Cejuela et al., 2018] Juan Miguel Cejuela, Shrikant Vinchurkar, Tatyana Goldberg, Madhukar Sollepura Prabhu Shankar, Ashish Baghudana, Aleksandar Bojchevski, Carsten Uhlig, André Ofner, Pandu Raharja-Liu, Lars Juhl Jensen, et al. Loctext: relation extraction of protein localizations to assist database curation. *BMC bioinformatics*, 19(1):15, 2018.

[Cheng et al., 2017] Heng-Tze Cheng, Zakaria Haque, Lichan Hong, Mustafa Ispir, Clemens Mewald, Illia Polosukhin, Georgios Roumpos, D Sculley, Jamie Smith, David Soergel, et al. Tensorflow estimators: Managing simplicity vs. flexibility in high-level machine learning frameworks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1763–1771. ACM, 2017.

[Coelho, 2019] Madelaine E Coelho. Rape myths in digital spaces: An analysis of high-profile sexual assault cases on twitter. 2019.

[Culotta et al., 2015] Aron Culotta, Nirmal Ravi Kumar, and Jennifer Cutler. Predicting the demographics of twitter users from website traffic data. In *AAAI*, pages 72–78, 2015.

[Dash and Liu, 1997] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(1-4):131–156, 1997.

[Dean et al., 2012] Jeffrey Dean, Greg S Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Quoc V Le, and Mark Z Mao. Marcâaurelioranzato, andrew senior, paul tucker, ke yang, andrew y. ng. large scale distributed deep networks. NIPS, 2012.

[Devries et al., 2013] Karen M Devries, Joelle YT Mak, Claudia Garcia-Moreno, Max Petzold, James C Child, Gail Falder, Stephen Lim, Loraine J Bacchus, Rebecca E Engell, Lisa Rosenfeld, et al. The global prevalence of intimate partner violence against women. *Science*, 340(6140):1527–1528, 2013.

[dos Santos and Gatti, 2014] Cicero dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[Dunkle et al., 2004] Kristin L Dunkle, Rachel K Jewkes, Heather C Brown, Mieko Yoshihama, Glenda E Gray, James A McIntyre, and Siobán D Harlow. Prevalence and patterns of gender-based violence and revictimization among women attending antenatal clinics in soweto, south africa. *American journal of epidemiology*, 160(3):230–239, 2004.

[Edwards et al., 2014] Paul K Edwards, Joe O'Mahoney, and Steve Vincent. *Studying organizations using critical realism: A practical guide*. OUP Oxford, 2014.

[Fersini et al., 2014] Elisabetta Fersini, Enza Messina, and Federico Alberto Pozzi. Sentiment analysis: Bayesian ensemble learning. *Decision support systems*, 68:26–38, 2014.

[Fink et al., 2012] Clayton Fink, Jonathon Kopecky, and Maksym Morawski. Inferring gender from the content of tweets: A region specific example. In *ICWSM*, 2012.

[Flood, 2019] Michael Flood. The problem: Men's violence against women. In *Engaging Men and Boys in Violence Prevention*, pages 11–38. Springer, 2019.

[Gilbert et al., 2014] Glen G Gilbert, Robin G Sawyer, and Elisa Beth McNeill. *Health education: Creating strategies for school & community health*. Jones & Bartlett Publishers, 2014.

[Go et al., 2009] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

[Graber and Dunaway, 2017] Doris A Graber and Johanna Dunaway. *Mass media and American politics*. Cq Press, 2017.

[Greenwood et al., 2016] Shannon Greenwood, Andrew Perrin, and Maeve Duggan. Social media update 2016. *Pew Research Center*, 11(2), 2016.

[Habermas, 2015] Jürgen Habermas. *Communication and the Evolution of Society*. John Wiley & Sons, 2015.

[Hamel, 2009] Lutz Hamel. Model assessment with roc curves. In *Encyclopedia of Data Warehousing and Mining, Second Edition*, pages 1316–1323. IGI Global, 2009.

[Hamilton et al., 2018] James Hamilton, Manuel Gonzalez Berges, Jean-Charles Tournier, and Brad Schofield. Jacow: Scada statistics monitoring using the elastic stack (elasticsearch, logstash, kibana). 2018.

[Heck and Huang, 2014] Larry Heck and Hongzhao Huang. Deep learning of knowledge graph embeddings for semantic parsing of twitter dialogs. In *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 597–601. IEEE, 2014.

[Holzman, 1996] Clare G Holzman. Counseling adult women rape survivors: Issues of race, ethnicity, and class. *Women & Therapy*, 19(2):47–62, 1996.

[Imuede et al., 2020] Jude Imuede, Mpho Raborife, and Pravesh Ranchod. Sentiment analysis as an indicator to evaluate gender disparity on sexual violence tweets in south africa. In *2020 International SAUPEC/RobMech/PRASA Conference*, pages 1–6. IEEE, 2020.

[Jewkes and Morrell, 2010] Rachel Jewkes and Robert Morrell. Gender and sexuality: emerging perspectives from the heterosexual epidemic in south africa and implications for hiv risk and prevention. *Journal of the International AIDS society*, 13(1):6, 2010.

[Jewkes et al., 2002] Rachel Jewkes, Jonathan Levin, and Loveday Penn-Kekana. Risk factors for domestic violence: findings from a south african cross-sectional study. *Social science & medicine*, 55(9):1603–1617, 2002.

[Jia et al., 2014] Yaoqi Jia, Xinshu Dong, Zhenkai Liang, and Prateek Saxena. I know where you've been: Geo-inference attacks via the browser cache. *IEEE Internet Computing*, 19(1):44–53, 2014.

[Jurgens et al., 2015] David Jurgens, Tyler Finethy, James McCorriston, Yi Tian Xu, and Derek Ruths. Geolocation prediction in twitter using social networks: A critical analysis and review of current practice. In *Ninth International AAAI Conference on Web and Social Media*, 2015.

[KaggleGenderClassification] KaggleGenderClassification. Twitter user gender classification — kaggle. https://www.kaggle.com/crowdflower/twitter-user-gender-classification. (Accessed on 01/07/2020).

[Kelly, 2013] Liz Kelly. *Surviving sexual violence*. John Wiley & Sons, 2013.

[Khan et al., 2018] Imran Khan, SK Naqvi, Mansaf Alam, and SNA Rizvi. A framework for twitter data analysis. In *Big Data Analytics*, pages 297–303. Springer, 2018.

[Kumar et al., 2014] Shamanth Kumar, Fred Morstatter, and Huan Liu. *Twitter data analytics*. Springer, 2014.

[Lahmadi et al., 2015] Abdelkader Lahmadi, Frederic Beck, Eric Finickel, and Olivier Festor. A platform for the analysis and visualization of network flow data of android environments. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 1129–1130. IEEE, 2015.

[Lasorsa, 2012] Dominic Lasorsa. Transparency and other journalistic norms on twitter: The role of gender. *Journalism Studies*, 13(3):402–417, 2012.

[Lewis and Brown, 2001] HG Lewis and M Brown. A generalized confusion matrix for assessing area estimates from remotely sensed data. *International Journal of Remote Sensing*, 22(16):3223–3235, 2001.

[Longley et al., 2015] Paul A Longley, Muhammad Adnan, and Guy Lansley. The geotemporal demographics of twitter usage. *Environment and Planning A*, 47(2):465–484, 2015.

[Mabasa, 2009] Thandi Beatrice Mabasa. *Understanding and preventing rape: perceptions of police officers in inner city Johannesburg*. PhD thesis, 2009.

[Marquarding, 2018] Malte Marquarding. Australian square kilometre pathfinder-commissioning to operations. 2018.

[McGregor and Mourão, 2016] Shannon C McGregor and Rachel R Mourão. Talking politics on twitter: Gender, elections, and social networks. *Social Media+ Society*, 2(3):2056305116664218, 2016.

[Mendes et al., 2018] Kaitlynn Mendes, Jessica Ringrose, and Jessalynn Keller. # metoo and the promise and pitfalls of challenging rape culture through digital feminist activism. *European Journal of Women's Studies*, 25(2):236–246, 2018.

[Michel et al., 2012] Vincent Michel, Alexandre Gramfort, Gaël Varoquaux, Evelyn Eger, Christine Keribin, and Bertrand Thirion. A supervised clustering approach for fmri-based inference of brain states. *Pattern Recognition*, 45(6):2041–2049, 2012.

[Miller et al., 2012] Zachary Miller, Brian Dickinson, and Wei Hu. Gender prediction on twitter using stream algorithms with n-gram character features. *International Journal of Intelligence Science*, 2(04):143, 2012.

[Mislove et al., 2011] Alan Mislove, Sune Lehmann, Yong-Yeol Ahn, Jukka-Pekka Onnela, and J Niels Rosenquist. Understanding the demographics of twitter users. *ICWSM*, 11(5th):25, 2011.

[Miura et al., 2017] Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1260–1272, 2017.

[Morstatter et al., 2013] Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In *Seventh international AAAI conference on weblogs and social media*, 2013.

[Nagara et al., 2017] Keigo Nagara, Katsunori Aoki, Yutaka Matsubara, and Hiroaki Takada. Portable dos test tool for iot devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pages 57–58. ACM, 2017.

[Nguyen et al., 2016] Dat Tien Nguyen, Shafiq Joty, Muhammad Imran, Hassan Sajjad, and Prasenjit Mitra. Applications of online deep learning for crisis response using social media information. *arXiv preprint arXiv:1610.01030*, 2016.

[Nguyen et al., 2014] Dong Nguyen, Dolf Trieschnigg, A Seza Doğruöz, Rilana Gravel, Mariët Theune, Theo Meder, and Franciska De Jong. Why gender and age prediction from tweets is hard: Lessons from a crowdsourcing experiment. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1950–1961, 2014.

[Nielsen, 2011a] F. Å. Nielsen. Afinn, mar 2011a. URL http://www2.imm.dtu.dk/pubdb/p.php?6010.

[Nielsen, 2011b] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011b.

[Organization et al., 2012] World Health Organization et al. Understanding and addressing violence against women: Intimate partner violence. Technical report, World Health Organization, 2012.

[Ozenne et al., 2015] Brice Ozenne, Fabien Subtil, and Delphine Maucort-Boulch. The precision–recall curve overcame the optimism of the receiver operating characteristic curve in rare diseases. *Journal of clinical epidemiology*, 68(8):855–859, 2015.

[OâKeefe and Koprinska, 2009] Tim OâKeefe and Irena Koprinska. Feature selection and weighting methods in sentiment analysis. In *Proceedings of the 14th Australasian document computing symposium, Sydney*, pages 67–74. Citeseer, 2009.

[Pak and Paroubek, 2010] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[Pan and Yang, 2009] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[Pang and Lee, 2004] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[Pang et al., 2002] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[Patricia and Caputo, 2014] Novi Patricia and Barbara Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1442–1449, 2014.

[Pedregosa et al., 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[Pitcher and Bowley, 2002] Graeme J Pitcher and Douglas MG Bowley. Infant rape in south africa. *The Lancet*, 359(9303):274–275, 2002.

[Powers, 2011] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

[Rao et al., 2010] Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM, 2010.

[Reips and Garaizar, 2011] Ulf-Dietrich Reips and Pablo Garaizar. Mining twitter: A source for psychological wisdom of the crowds. *Behavior research methods*, 43(3):635, 2011.

[Ren et al., 2016] Yafeng Ren, Yue Zhang, Meishan Zhang, and Donghong Ji. Context-sensitive twitter sentiment classification using neural network. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Severyn and Moschitti, 2015] Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM, 2015.

[Shah et al., 2018] Neel Shah, Darryl Willick, and Vijay Mago. A framework for social media data analytics using elasticsearch and kibana. *Wireless Networks*, pages 1–9, 2018.

[Sola and Sevilla, 1997] J Sola and Joaquin Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science*, 44(3):1464–1468, 1997.

[StatsSA, 2018] Statistics South Africa StatsSA. *Crime Against Women in South Africa*. 2018.

[Taboada et al., 2011] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[Turney, 2002] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[Vogel and Jiang, 2019] Inna Vogel and Peter Jiang. Bot and gender identification in twitter using word and character n-grams. 2019.

[Volkova et al., 2016] Svitlana Volkova, Yoram Bachrach, and Benjamin Van Durme. Mining user interests to predict perceived psycho-demographic traits on twitter. In *2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 36–43. IEEE, 2016.

[Wagner III and Hollenbeck, 2014] John A Wagner III and John R Hollenbeck. *Organizational behavior: Securing competitive advantage*. Routledge, 2014.

[Wojcicki, 2002] Janet Maia Wojcicki. " she drank his money": survival sex and the problem of violence in taverns in gauteng province, south africa. *Medical anthropology quarterly*, 16(3):267–293, 2002.

[Wongsuphasawat et al., 2018] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mané, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. Visualizing dataflow graphs of deep learning models in tensorflow. *IEEE transactions on visualization and computer graphics*, 24(1):1–12, 2018.

[Wu et al., 2008] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.

[Yang and Wu, 2006] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.

[Yuan et al., 2016] Shuhan Yuan, Xintao Wu, and Yang Xiang. A two phase deep learning model for identifying discrimination from tweets. In *EDBT*, pages 696–697, 2016.