



Strathmore
UNIVERSITY

Strathmore University
SU+ @ Strathmore
University Library

Electronic Theses and Dissertations

2018

Open source intelligence gathering for hate speech in Kenya

Banchale G. Adhi
Faculty of Information Technology (FIT)
Strathmore University

Follow this and additional works at <https://su-plus.strathmore.edu/handle/11071/5980>

Recommended Citation

Adhi, B. G. (2018). *Open source intelligence gathering for hate speech in Kenya*

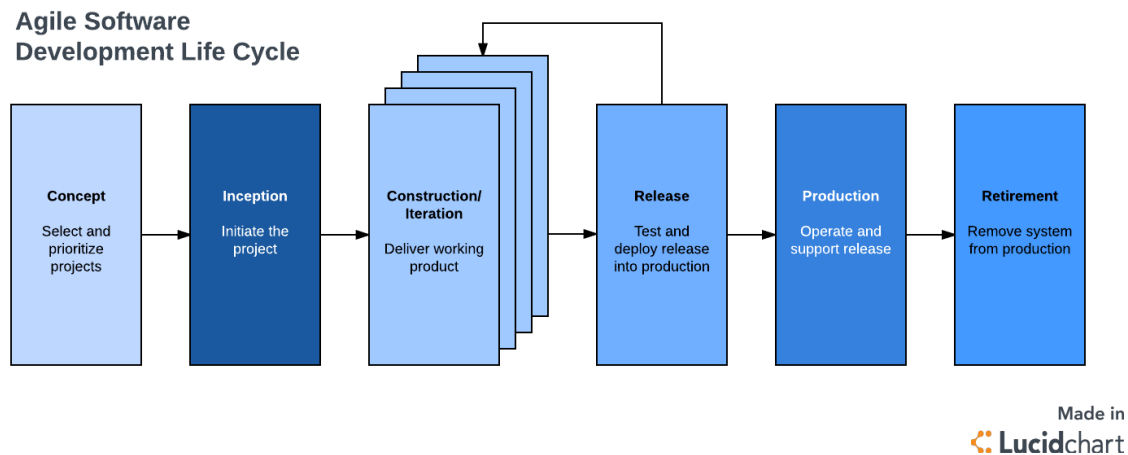
(Thesis). Strathmore University. Retrieved from <https://su->

[plus.strathmore.edu/handle/11071/5980](https://su-plus.strathmore.edu/handle/11071/5980)

TABLE OF CONTENTS

DECLARATION.....	ii
ABSTRACT.....	iii
LIST OF ABBREVIATIONS	viii
DEFINITION OF TERMS.....	ix
ACKNOWLEDGEMENT.....	xi
DEDICATION.....	xii
CHAPTER ONE: INTRODUCTION.....	13
1.1 Background	13
1.2 Social Media	15
1.3 Open Source Intelligence.....	16
1.4 Statement of the Problem.....	17
1.5 Research Objectives.....	18
1.6 Research Questions.....	18
1.7 Relevance of the Research.....	19
1.8 Scope and Limitations	19
CHAPTER TWO: LITERATURE REVIEW.....	20
2.1 Introduction.....	20
2.2 Sentiment Analysis Algorithms	20
2.2.1 Lexicon Algorithms	20
2.2.2 Machine Learning Algorithms (MLA)	21
2.3 Understanding Social Media in Kenyan Perspective.....	23
2.4 Challenges Regulating Social Media	25
2.5 Developing a Speech Analysis Tool	25
2.6 Existing Tools in Hate Speech Detection	27
2.6.1 Perspective API.....	27
2.6.2 Spice Hate Speech Detection	28
2.6.3 Hate Speech Blocker.....	28
2.6.4 Umati Online Monitoring Project in Kenya	28
2.6.5 Uchaguzi Online Monitoring Project in Kenya.....	29
2.6 Conclusions.....	30
CHAPTER THREE: RESEARCH METHODOLOGY	32
3.1 Challenge Identification	32
3.2 Research Design	32
3.3 Data Collection Methods	33
3.3.1 Observation	33
3.3.2 Questionnaires	33
3.4 Data Classification and Analysis	34
3.5 Implementation of the System	34
3.6 Overview of Agile Software Development Methodology.....	35
3.7 Agile Software Development Life Cycle (SDLC)	35
3.7.1 Concept.....	36
3.7.2 Inception	36

Figure 3.1: The Stages of the Agile Software Development Life Cycle



Source: The Stages of the Agile Software Development Life Cycle.

Retrieved from <https://www.lucidchart.com/blog/agile-software-development-life-cycle>

3.7.1 Concept

It is the first stage of Agile SDLC. With close observation, it was identified that there were problems in the hate speech detection process, from detection to response of an incidence. The main problem identified was that manual detection was employed. With this in mind, consultations were made and it was agreed upon that there was a great need for a tool to automate the detection process. This research focused on the development of automated hate speech detection tool, which would facilitate hate speech detection process in Kenya in a more suitable manner guaranteeing timely response.

3.7.2 Inception

The researcher worked closely with NCIC hate speech monitoring analysts during the tool's development to determine the requirements that were needed during the development of this

- Hatzivassiloglou, V., Wiebe, J. (2000). Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In: *Proceedings of the 18th International Conference on Computational Linguistics*, New Brunswick, NJ
- Hitz, L., Blackburn, B. (2017). *The State of Social Marketing 2017 Annual Report*. Retrieved from https://get.simplymeasured.com/rs/135-YGJ-288/images/SM_StateOfSocial-2017.pdf
- Joachims, T. (1998). Text Categorization with Support Vector Machine: Learning with Many Relevant Features. *Proceedings of the 10th European Conference on Machine Learning* (pp. 137-142). London: Springer-Verlag.
- Kaplan, A. M., & Haenlein, M. (2011). The early bird catches the news: Nine things you should know about micro-blogging. *Business horizons*, 54(2).
- Kennedy, A., Inkpen, D. (2016). Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters, *Computational Intelligence*.
- Kinnunen, T. (2017). Hate speech detection. Retrieved from <https://futuraice.com/blog/hate-speech-detection>
- Lichterman, J. (2017). This tool from Google parent Alphabet tries to tackle “toxic” comments through machine learning. Retrieved from <http://www.niemanlab.org/2017/02/this-tool-from-google-parent-alphabet-tries-to-tackle-toxic-comments-through-machine-learning/>
- Liombart, R. Ò., & Duran, C. J. (2017). Using machine learning techniques for sentiment analysis.
- Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1), 1-167.
- Maloba, W. (2013). *Use of regular expressions for multilingual detection of hate speech in Kenya*. (Published MSc. thesis) MMTI Theses and Dissertations (2013). (2198)
- Martini, B., Do, Q., & Raymond Choo, K. K. (2016). Digital forensics in the cloud era: The decline of passwords and the need for legal reform. *Trends & Issues in Crime & Criminal Justice*, (512).
- Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60).
- Morales D. R. (2018). Django-comments-xtD Documentation. Release 2.0.1. Retrieved from <https://media.readthedocs.org/pdf/django-comments-xtD/latest/django-comments-xtD.pdf>
- Mutahi, P., & Kimari, B. (2017). *The Impact of Social Media and Digital Technology on Electoral Violence in Kenya*. IDS.
- National Council for Law Reporting. (2008). National Cohesion and Integration Act. Nairobi.
- Ogada, K., Mwangi, W., & Cheruiyot, W. (2015). N-gram Based Text Categorization Method for Improved Data Mining. *Journal of Information Engineering and Applications*, 5(8), 35-43
- Omenya, R. (2013) *Uchaguzi Kenya 2013: Monitoring & Evaluation*. iHub Research and HIVOS. Retrieved from http://www.ihub.co.ke/ihubresearch/jb_UchaguziMEFinalReportpdf2013-7-5-14-24-09.pdf
- Peng, F. (2003). Augmenting Naïve Bayes Classifiers with Statistical. *University of Massachusetts, Computer Science Department Faculty Publication Series*.
Published: Ondingi O Nyambane ‘prosecuting hate speech in Kenya ‘published dissertation, University of Nairobi, 2012 Ondingi O Nyambane.
- Restricted use of Twitter APIs. (2018). Retrieved from <https://developer.Twitter.com/en/developer-terms/more-on-restricted-use-cases>

- Rudkowsky, E., Haselmayer, M., Wastian, M., Jenny, M., Emrich, Š., & Sedlmair, M. (2018). More than Bags of Words: Sentiment Analysis with Word Embeddings. *Communication Methods and Measures*, 12(2-3), 140-157.
- Sambuli, N., Morara, F., & Mahihu, C. (2013). *Monitoring Online Dangerous Speech in Kenya*. Nairobi: Umati.
- Satapathy, S. C., Govardhan, A., Raju, K. S., & Mandal, J. K. (Eds.). (2014). Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India (CSI) (Vol. 1). Springer.
- Silva, L. A., Mondal, M., Correa, D., Benevenuto, F., & Weber, I. (2016, March). Analysing the Targets of Hate in Online Social Media. In ICWSM (pp. 687-690).
- Skyrme, D. (2007). Knowledge networking: Creating the collaborative enterprise. Routledge.
- Social@Ogilvy. (2015). *Social Media in Africa*. Retrieved from https://social.ogilvy.com/wp-content/uploads/Social-Media-in-Africa_Infographic.pdf
- Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010, July). Short text classification in Twitter to improve information filtering. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval* (pp. 841-842). ACM.
- Standard. (2016, December 05). *How app assists parents manage child's phone*. Retrieved from <https://www.standardmedia.co.ke/business/article/2000225838/how-app-assists-parents-manage-child-s-phone>
- State cracks down on 176 social media accounts over hate speech. (2017, July 28). Retrieved from https://www.the-star.co.ke/news/2017/07/28/state-cracks-down-on-176-social-media-accounts-over-hate-speech_c1606015
- Strachan, A. L. (2014). Interventions to counter hate speech. GSDRC Applied Research Services, 23.
- Taylor, M., Haggerty, J., Gresty, D., Almond, P., & Berry, T. (2014). Forensic investigation of social networking applications. *Network Security*, 2014(11), 9-16.
- Thakkar, H., and Patel D. (2015) Approaches for Sentiment Analysis on Social media: A State-of-Art study.
- The Bloggers Association of Kenya (BAKE). (2015). *The State of Blogging & Social Media in Kenya 2015 Report* [Ebook]. Nairobi. Retrieved from <http://www.monitor.co.ke/wp-content/uploads/2015/06/The-State-of-Blogging-and-Social-Media-in-Kenya-2015-report.pdf>
- The Stages of the Agile Software Development Life Cycle. (2017, December 01). Retrieved March 14, 2018, Retrieved from <https://www.lucidchart.com/blog/agile-software-development-life-cycle>
- Turney, P. D., (2002) "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews classification of reviews", *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 417-424.
- Understanding the Agile Software Development Lifecycle and Process Workflow. (2017, October 19). Retrieved March 14, 2018, Retrieved from <https://www.smartsheet.com/understanding-agile-software-development-lifecycle-and-process-workflow>
- Wang, H., Can, D., Kazemzadeh, A., Bar, F., & Narayanan, S. (2012, July). A system for real-time Twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations* (pp. 115-120). Association for Computational Linguistics.

- Wiebe, J., Wilson, T., & Cardie, C. (2005). Annotating expressions of opinions and emotions in language resources and evaluation, 39(2-3), 165-210.
- Yahyaoui, M. (2001). Toward an Arabic web page classifier, Master project. AUI.
- Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems.



APPENDIX A: Interview Guide

**STRATHMORE UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGY
MASTER OF SCIENCE IN INFORMATION SYSTEMS SECURITY**

Research Questionnaire

I am a graduate student at the Strathmore University, Faculty of Information Technology. I am conducting a research in partial fulfilment of a Masters in Information System Security (MISS). My research aims at developing an Open Source Intelligence Gathering tool for Hate Speech in Kenya.

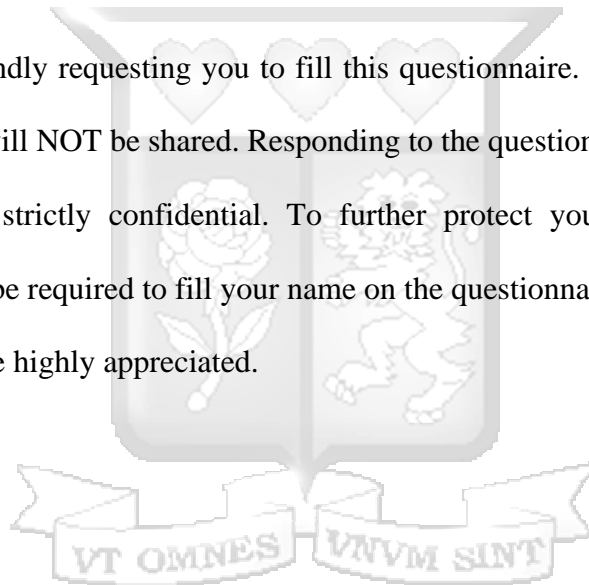
I am therefore kindly requesting you to fill this questionnaire. This survey is strictly for academic purposes and will NOT be shared. Responding to the questionnaire is voluntary and the responses will be kept strictly confidential. To further protect your opinions and enhance anonymity, you will not be required to fill your name on the questionnaire.

Your co-operation will be highly appreciated.

Yours Faithfully:

.....

Banchale A. Gufu



Date.....

Questionnaire NO.....

The following interview guide was used to in a personal interview with staff members of the NCIC to find out the challenges faced in monitoring hate speech on social media.

Questions

1. Do you have any systems in place to help monitor hate speech on social media?

.....
.....
.....
.....

2. Do you currently monitor hate speech on social media?

a. If yes, how do you monitor hate speech on social media?

.....
.....
.....
.....

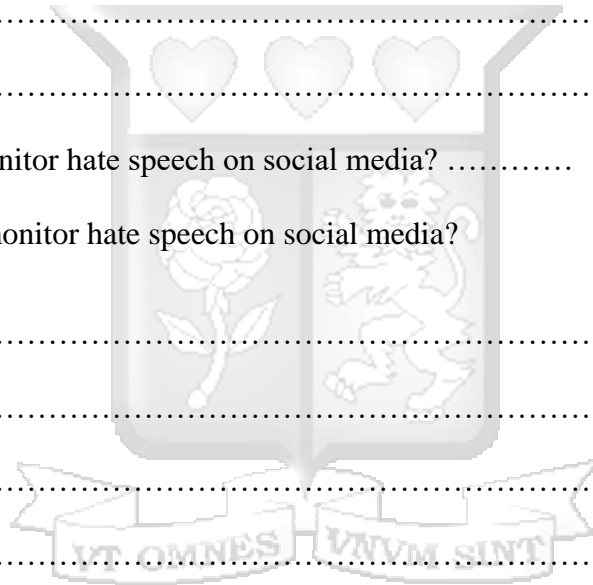
b. If not, why?

.....
.....
.....

3. Which social media sites do you monitor?

1.

2.



3.

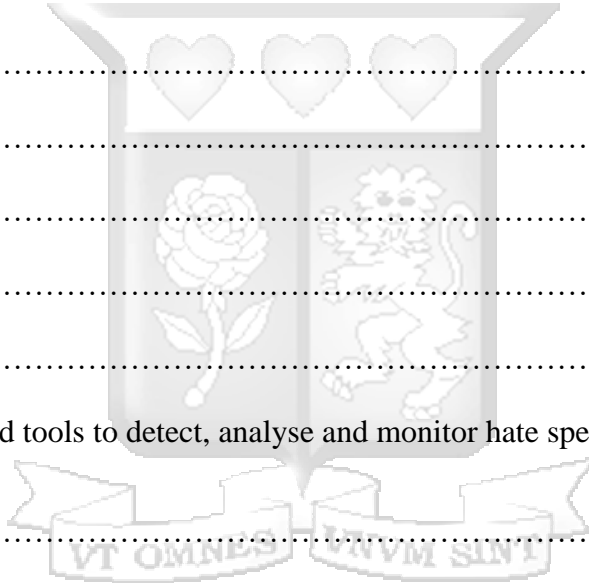
4.

others.....

3. How often do you monitor the social media sites for hate speech?

.....

4. What challenges do you face while monitoring hate speech on social media?



.....
.....
.....
.....
.....

5. Do you use automated tools to detect, analyse and monitor hate speech on social media?

.....
.....

6. Have you identified any gaps in the tools you use for hate speech detection, analysis and monitoring?

a. If yes, which are they?

.....
.....
.....
.....

7. How do you analyse collected data from social media?

.....

.....

.....

8. What challenges do you face while analysing collected data from social media?

.....

.....

.....

9. How do you deal with the multilingual nature of hate speech in Kenya on social media?

.....

.....

.....

10. Which are the most frequent terms found in hate speech text?

.....

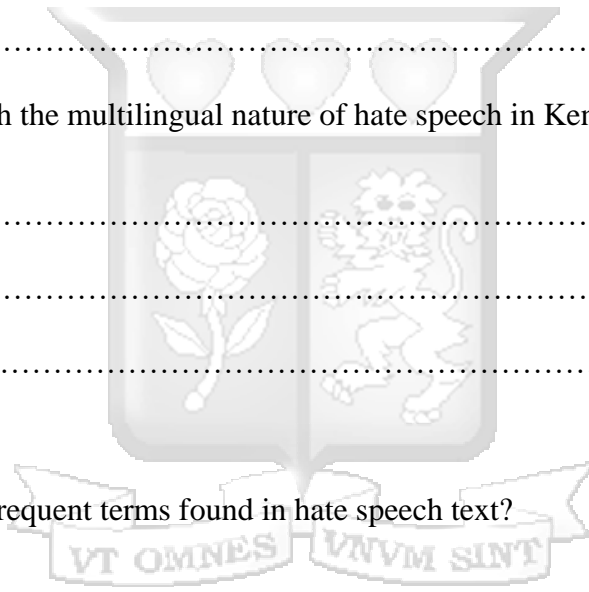
.....

.....

11. Which organisation(s) do you collaborate with in the detection, analysis and monitoring of hate speech on social media?

.....

.....



APPENDIX B: Python Program

a) General Python Source Code followed by the four processes as shown in the GUI

```
99 function main(){
100
101 OPTIONS=$(whiptail --title "OSINT GATHERING FOR HATE SPEECH" --menu "Select Option.. " 20 60 8 \
102 "1" "Configure Twitter API Keys" \
103 "2" "Collect Tweets" \
104 "3" "Clean Tweets" \
105 "4" "Analyse Tweets" \
106 "5" "Exit" 3>&1 1>&2 2>&3)
107
108 exitstatus=$?
109 exception_handler $exitstatus "[!] Failed To Start Menu"
110
111 if [ $OPTIONS = 1 ]; then
112     API_CONFIG
113
114     exitstatus=$?
115     exception_handler $exitstatus "[!] Failed To Configure KEYS"
116
117 elif [ $OPTIONS = 4 ]; then
118     ANALYZE_TWEETS
119
120     exitstatus=$?
121     exception_handler $exitstatus "[!] Failed To Analyze Tweets"
122
123 elif [ $OPTIONS = 2 ]; then
124     COLLECT_TWEETS
125
126     exitstatus=$?
127     exception_handler $exitstatus "[!] Failed To Collect Tweets"
128
129
```



b) Source Code of Twitter Credentials

```
1 ## Twitter credentials
2
3 consumer_key = 'PsnaQDBsgjBGF9eLR3LsCxhm2'
4 consumer_secret = 'uJwtkFHhD0kyybRJWi0rOZ1tYRxH0QIBfbnbZ3RS0FSKdeREiT'
5 access_token = '263544909-tWl0h6AE9yGDTxbl8JHmgumRKU2xJUeIaReiWHn'
6 access_token_secret = 'T52ZojWaTlYr4w9azHRJVA89Bf7dRumTd47dj8WGvNfnd'
7
8
```

c) Twitter Authentication and Twitter Streaming API Connection Source Code

```
#This handles Twitter authentication and the connection to Twitter Streaming API
auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

d) Tweets Collection Source Code

```
4
5 from tweepy.streaming import StreamListener
6 import tweepy.streaming
7 from tweepy import OAuthHandler
8 from tweepy import Stream
9
10 from api_config import *
11 import sqlite3
12
13 class CustomStreamListener(StreamListener):
14
15     def on_status(self, status):
16         db_file = 'Database/tweets.db'
17         con = sqlite3.connect(db_file)
18         #con.text_factory = str
19         con.text_factory = lambda x: unicode(x, 'utf-8', 'ignore')
20         cur = con.cursor()
21
22         tweet_date = status.created_at
23         try:
24             text = status.extended_tweet["full_text"]
25         except AttributeError:
26             text = status.text
27         tweet_text = text.encode('utf-8').translate(None, '!.?')
28
29         print "%s\t%s" % (tweet_date, tweet_text)
30         cur.execute("INSERT INTO raw_tweets(date, tweets) VALUES (?, ?)", (tweet_date, tweet_text))
31         con.commit()
32         con.close()
33
34 #This handles Twitter authentication and the connection to Twitter Streaming API
35 auth = OAuthHandler(consumer_key, consumer_secret)
36 auth.set_access_token(access_token, access_token_secret)
37
38 #This line filter Twitter Streams to capture data by the keywords
39 streaming_api = tweepy.streaming.Stream(auth, CustomStreamListener(),
40                                         timeout=60, tweet_mode='extended')
41 streaming_api.filter(track=['alshābab', 'mjinga', 'jeuri', 'kikuyu', 'jalu',
42                             'handcheque', 'raila', 'therealraila', 'uhuru'].asvnc=True)
```

e) Tweets Cleaning Code

```
1 import re
2 import sqlite3 as lite
3
4 db_file = 'Database/tweets.db'
5 con = lite.connect(db_file)
6 con.text_factory = lambda x: unicode(x, 'utf-8', 'ignore')
7 cur = con.cursor()
8
9 def export(filename, data, p):
10     with open(filename, p) as output:
11         for line in data:
12             output.write(line)
13
14 def cleanTweets(tweet):
15     cleantweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', 'URL', cleantweet)
16     cleantweet = re.sub('@[^\s]+', 'AT_USER', cleantweet)
17     cleantweet = re.sub('[\s]+', ' ', cleantweet)
18     cleantweet = re.sub(r'#([^\s]+)', r'\1', cleantweet)
19     cleantweet = cleantweet.strip('\n')
20     cleantweet = cleantweet.lower()
21     return cleantweet
22
23 all_cleaned_tweets = []
24
25 for row in cur.execute('SELECT * FROM raw_tweets;'):
26     twit = row[2]
27     clean_tweet = cleanTweets(twit)
28
29     print(row[0], row[1], clean_tweet)
30
31     tweet = (row[0], row[1], clean_tweet)
32     all_cleaned_tweets.append(tweet)
33
34 for item in all_cleaned_tweets:
35     cur.execute("INSERT INTO cleaned_tweets(id, date, clean_tweet) VALUES (?, ?, ?)", (item[0], item[1], item[2]))
36     con.commit()
37
38 con.close()
```

f) Tweets Preprocessing Source Code

```
1 from classifier import *
2 import sqlite3 as lite
3
4 posDB = 'test_data/positive_test.txt'
5 negDB = 'test_data/negative_test.txt'
6 posScore = []
7 negScore = []
8 neuScore = []
9
10 def plolarityCount(ourscore):
11     if ourscore > 0:
12         print "positive tweet : " + str(ourscore)
13         posScore.append(ourscore)
14     elif ourscore < 0:
15         print "Negative tweet : " + str(score)
16         negScore.append(ourscore)
17     else:
18         print "Neutral : " + str(score)
19         neuScore.append(ourscore)
20
21 def processTweet(tweet):
22     tweet = re.sub('((www\.[^\s]+)|(https?:\/\/[^\s]+))', 'URL', tweet)
23     tweet = re.sub('@[^\s]+', 'AT_USER', tweet)
24     tweet = re.sub('[\s]+', ' ', tweet)
25     tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
26     tweet = tweet.strip('\')
27     tweet = tweet.lower()
28     return tweet
29
```

VT OMNES VIVVM SINTE

g) Tweets Classification Code

```
1 from classifier import *
2 import sqlite3 as lite
3
4 db_file = 'Database/tweets.db'
5 con = lite.connect(db_file)
6 con.text_factory = lambda x: unicode(x, 'utf-8', 'ignore')
7 cur = con.cursor()
8
9
10 def classify_tweet(tweet):
11     return classifier.classify(extract_features(tweet.split()))
12
13 all_analysed_tweets = []
14
15 for row in cur.execute('SELECT * FROM cleaned_tweets;'):
16     twit = row[2]
17     sentiment = classify_tweet(twit)
18
19     if (sentiment != "neg") and (sentiment != "pos"):
20         sentiment = "neutral"
21
22     polarised_tweet = (row[0], row[1], row[2], sentiment)
23     all_analysed_tweets.append(polarised_tweet)
24
25
26 for item in all_analysed_tweets:
27     print item
28     cur.execute("INSERT INTO analysed_tweets(id, date, analysed_tweet, polarity) VALUES (?, ?, ?, ?)",
29               (item[0], item[1], item[2], item[3]))
30     con.commit()
31
32
33 con.close()
```

h) Training Data

```
1 # coding=utf-8
2 # Authour - B. Gufu
3 # @ilabafrica, Strathmore University
4
5 import os
6
7 def getTrainData():
8     positives, negatives, traindata = [], [], []
9     for filename in os.listdir("training_set"):
10         if filename == "positive_tweets.txt":
11             with open('training_set/'+filename) as f:
12                 positives = [(tweet, 'pos') for tweet in f.readlines()]
13         if filename == "negative_tweets.txt":
14             with open('training_set/'+filename) as f:
15                 negatives = [(tweet, 'neg') for tweet in f.readlines()]
16
17     for (words, sentiment) in negatives + positives:
18         words_filtered = [e for e in words.split() if len(e) > 2]
19         traindata.append((words_filtered, sentiment))
20
21     return traindata
22
```

i) Analyses Source Code

i. Part 1

```
1 from classifier import *
2 import sqlite3 as lite
3
4 posDB = 'test_data/positive_test.txt'
5 negDB = 'test_data/negative_test.txt'
6 posScore = []
7 negScore = []
8 neuScore = []
9
10 def plolarityCount(ourscore):
11     if ourscore > 0:
12         print "positive tweet : " + str(ourscore)
13         posScore.append(ourscore)
14     elif ourscore < 0:
15         print "Negative tweet : " + str(score)
16         negScore.append(ourscore)
17     else:
18         print "Neutral : " + str(score)
19         neuScore.append(ourscore)
20
21 def processTweet(tweet):
22     tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', 'URL', tweet)
23     tweet = re.sub('@[^\s]+', 'AT_USER', tweet)
24     tweet = re.sub('[\s]+', ' ', tweet)
25     tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
26     tweet = tweet.strip('\')
27     tweet = tweet.lower()
28     return tweet
29
```

ii. Part 2

```
21 def processTweet(tweet):
22     tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))', 'URL', tweet)
23     tweet = re.sub('@[^\s]+', 'AT_USER', tweet)
24     tweet = re.sub('[\s]+', ' ', tweet)
25     tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
26     tweet = tweet.strip('\')
27     tweet = tweet.lower()
28     return tweet
29
30 def classify_tweet(tweet):
31     return classifier.classify(extract_features(tweet.split()))
32
33 pos = open(posDB, 'r')
34 pos = pos.read()
35 negs = open(posDB, 'r')
36 negs = pos.read()
37
38 for line in pos:
39     ans = classify_tweet(line)
40     plolarityCount(ans)
41
42 for line in pos:
43     ans = classify_tweet(line)
44     plolarityCount(ans)
45
```