

**CSE 4022 – NATURAL LANGUAGE PROCESSING**

***J COMPONENT PROJECT DOCUMENT***

***FALL 2020-2021***

**GROUP/ TEAM PROJECT**

Navdeep Chawla 18BCE0458

Amaatye Mahajan 18BCE0918

Guneshwar Singh Manhas 18BCE0960

Abhishek Dhall 18BCE2278

UNDER THE GUIDANCE OF

PROF. SHARMILA BANU

SCOPE

**Introduction**

**Aim**

**Abstract**

Modern technological era has reshaped traditional lifestyle in several domains. The medium of publishing news and events has become faster with the advancement of Information Technology (IT). IT has also been flooded with immense amounts of data, which is being published every minute of every day, by millions of users, in the shape of comments, blogs, news sharing through blogs, social media micro-blogging websites and many more. Manual traversal of such huge data is a challenging job; thus, sophisticated methods are acquired to perform this task automatically and efficiently. News reports events that comprise of emotions-good, bad, neutral. Sentiment analysis is utilised to investigate human emotions (i.e., sentiments) present in textual information. This paper presents a lexicon-based approach for sentiment analysis of news articles. The experiments have been performed on BBC news dataset, which expresses the applicability and validation of the adopted approach.

**Introduction**

With the emergence of the Internet, web and mobile technologies, people have changed their way of consuming news. Traditional physical newspapers and magazines have been replaced by virtual online versions like online news and weblogs. Readers are more inclined to use online sources of news mainly due to two key features: interactivity and immediacy. In this day and age, people want to consume as much news, from as many sources, as they possibly can, on matters that are important to them or matters that catch their attention. Interactivity refers to the inherent tendency depicted by the masses that makes them consume news of their interest. Immediacy is a feature that represents the need of people to be informed about news with no delay in time. The world we live in and the technology we are accustomed to, allows people to benefit from these features by providing them instant news on events as they happen in real-time. Online news websites have developed effective strategies to draw peoples’ attention. Online news expresses opinions regarding news entities, which may comprise of people, places or even things, while reporting on events that have recently occurred. For this reason, interactive emotion rating services are offered by various channels of several news websites, i.e., news can be positive, negative or neutral. Sentiment Analysis is a way of finding out the polarity or strength of the opinion (positive or negative) that is expressed in written text, in the case of this paper –a news article. Manual labelling of sentiment words is a time-consuming process. We use the common process to automate the process of sentiment analysis in this project, that is based on approaches of machine learning.

**System Requirements**

**Recommended Operating Systems**

* **Windows:**7 or newer
* **MAC:** OS X v10.7 or higher
* **Linux:** Ubuntu

**Hardware Requirements**

* Processor: Minimum 1 GHz; Recommended 2GHz or more
* Hard Drive: Minimum 2 GB; Recommended 64 GB or more
* Memory (RAM): Minimum 1 GB; Recommended 4 GB or above

**Methodology**

The methodology used for sentiment analysis of news articles in this paper is based on the Lexicon-based approach. Sentiment analysis can generally be carried out using supervised or unsupervised approaches. A supervised approach comprises of a set of labeled training data that is used to build a classification model with the intent of using this model to classify new data for which labels are not present. Unsupervised or Lexicon-based approaches to sentiment analysis do not require any training data. In this approach, the sentiments conveyed by a word are inferred on grounds of the polarity of the word. In case of a sentence or a document, the polarities of the individual words that compose the document collectively convey the sentiment of the sentence or the document. Thus the polarity of a sentence is the accumulative total (sum) of polarities of the individual words (or phrases) in the sentence.

This approach utilises some predefined lists of words such that each word in the list is associated with a specific sentiment. Further this approach can use the followingmethods:

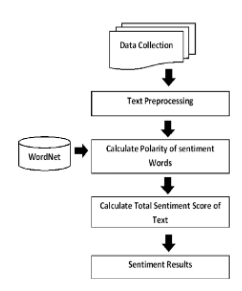
* 1. Dictionary-based methods: in these methods lexicon dictionary is used in order to find out the positive opinion words and negative opinion words.

Sentiment analysis can be done on document level, sentence level, word level or phrase level. This paper explores sentiment analysis on the document level. Similar to , this research identifies whether the documents new articles expressed opinions are positive, negative or neutral. The dictionary based approach has been used for sentiment analysis of news articlesutilisingthewordNetlexicaldictionary.Theexperimentforthis

research was carried out using the Rapid Miner tool.

The methodology for this experiment has been presented in the figure below. The methodology comprised of 5 steps, starting with data collection. The BBC News dataset has been used for this project. The next step was preprocessing the collected data in order to reduce inconsistencies in the dataset. The polarity of the words in the collected news articles was computed next using the wordNet lexical dictionary.

The steps have been explained in detail below.



**Data Collection**

The BBC News dataset was utilised for this experiment. The dataset is available online at http://mlg.ucd.ie/datasets/bbc.html.This data set comprises of a total of 2225 documents that comprise of news articles reported on the BBC news website of a year . The news stories belongto5 (five) topical areas. The dataset comprises of the following class labels: business, entertainment, politics, sport, and tech.

**Text Pre-processing**

News articles in the data set were preprocessed. Preprocessing is a necessary step to clean text (lessen noise of text) and to reduce inconsistencies from it so that this cleansed data can more effectively be utilised in text mining or sentiment analysis task. The entire preprocessing task was carried out using the Rapid miner tool which provides a vast set of operators for preprocessing tasks. The first preprocessing task was tokenising the text in news articles into a set of tokens by using the “Tokenize” operator. Tokenising breaks a sequence of sentences (combination of strings) into individual components such as words, phrases or symbols which are termed tokens. Apart from individual words and phrases, tokens can even comprise of entire sentences. During tokenisation some characters, such as punctuation marks, are discarded. After tokenisation, the text of the entire documents was changed to a lower case format using the “Transform cases” operator. Stop words from the text were removed using “filter stop word (English)” operator. The next task was reducing inflected or derived words through a process called Stemming. Stemming of words was done using the “stem (wordNet)” operator

**Calculating Polarity of Sentiment of Sentiment of words**

After preprocessing, the statistical technique known as Term Frequency-Inverse Document Frequency (TF-IDF) has been used. In TF-IDF term frequency is counted . According to this technique words that occur frequently in a document are considered important and a weight is given to these words. Using TF-IDF important words or terms in a document were identified and assigned a weightage according to the occurrence of various words in the news article. After identification of important words, a dictionary has been used for assigning sentiment score to the discovered

words. The WordNet dictionary, which is also known as a lexical database for English language, has been used in this experiment. WordNet contains more than 118,000 different word forms and more than 90,000 different word senses. WordNet provides accurate results to find opinion words in a given text and to give sentiment score to them.

**Calculate Total Sentiment Score**

According to the principle of document level sentiment analysis, each individual document is tagged with its respective polarity. This is generally done by finding polarities of each individual words/phrases and sentences and combining them to predict the polarity of whole document. Treating each new article as a document, the sentiment conveyed in the article has been computed by combining polarities of individual words/phrases and sentences in news articles. The sentiment score of whole news article has been calculated using the “extract sentiment” operator. This operator provides final results about sentiments: text having a sentiment score of -1 is considered negative and text having a sentiment score of +1 is positive. This operator provided accurate results by using SentiWordNet 3.0.0 dictionary which is actually an extension of the wordNet dictionary. WordNet and SentiWordNet are connected by Synset IDs. Also by using Score sentiment function based on WordNet and SentiWordNet dictionary, total sentiment score of news article was calculated.

**Sentiment Results**

News articles were classified in to positive, negative and neutral classes by looking at their total sentiment score. News articles sentiment was then calculated as the average value of total word sentiments.

**CODE AND OUTPUT**

**CODE**

**Train.py**

1. import nltk
2. import string
3. import random
4. from nltk.corpus import stopwords
5. from nltk.tokenize import word\_tokenize,sent\_tokenize
6. from nltk.classify.scikitlearn import SklearnClassifier
7. from sklearn.naive\_bayes import MultinomialNB, GaussianNB, BernoulliNB
8. from sklearn.linear\_model import LogisticRegression,SGDClassifier
9. from sklearn.svm import SVC,LinearSVC,NuSVC
10. import pickle
11. from nltk.classify import ClassifierI
12. from statistics import mode
13. import re
14. stop\_words = set(stopwords.words("english"))
15. class voteclassifier (ClassifierI):
16. def \_\_init\_\_(self, \*classifiers):
17. self.\_classifiers=classifiers
18. def classify(self, features):
19. votes = []
20. for c in self.\_classifiers:
21. v = c.classify(features)
22. votes.append(v)
23. return mode(votes)
24. def confidence(self, features):
25. votes = []
26. for c in self.\_classifiers:
27. v = c.classify(features)
28. votes.append(v)
29. choice\_votes = votes.count(mode(votes))
30. conf = choice\_votes/len(votes)
31. return conf
32. with open("politic.pickle", 'rb') as dic:
33. business = pickle.load(dic)
34. doc = []
35. pos = []
36. neg = []
37. pos\_words = []
38. neg\_words = []
39. all\_words = []
40. for i in business:
41. if business[i]==1:
42. doc.append((i, "pos"))
43. pos.append(i)
44. elif business[i]==-1:
45. doc.append((i, "neg"))
46. neg.append(i)
47. for i in pos:
48. a=word\_tokenize(i)
49. for word in a:
50. word=word.lower()
51. if word not in stop\_words:
52. if word not in string.punctuation:
53. if word.isalpha():
54. if re.match("(g\w+W+)","hello")!="None":
55. pos\_words.append(word)
56. for i in neg:
57. a=word\_tokenize(i)
58. for word in a:
59. word=word.lower()
60. if word not in stop\_words:
61. if word not in string.punctuation:
62. if word.isalpha():
63. if re.match("(g\w+W+)","hello")!="None":
64. neg\_words.append(word)
65. for i in pos\_words:
66. all\_words.append(i)
67. for i in neg\_words:
68. all\_words.append(i)
69. all\_words=nltk.FreqDist(all\_words)
70. words=list(all\_words.keys())[:5000]
71. def find\_feat(document):
72. word=word\_tokenize(document)
73. feat={}
74. for w in words:
75. feat[w]=(w in word)
76. return feat
77. feature\_Sets=[(find\_feat(rev),category) for (rev,category) in doc]
78. random.shuffle(feature\_Sets)
79. train=feature\_Sets[:300]
80. test=feature\_Sets[300:]
81. classifier=nltk.NaiveBayesClassifier.train(train)
82. print("original naive bayes",nltk.classify.accuracy(classifier,test))
83. classifier.show\_most\_informative\_features(15)
84. mnb\_classifier=SklearnClassifier(MultinomialNB())
85. mnb\_classifier.train(train)
86. print("mnb\_classifier",nltk.classify.accuracy(mnb\_classifier,test))
87. bnb\_classifier=SklearnClassifier(BernoulliNB())
88. bnb\_classifier.train(train)
89. print("bnb\_classifier",nltk.classify.accuracy(bnb\_classifier,test))
90. logr\_classifier=SklearnClassifier(LogisticRegression())
91. logr\_classifier.train(train)
92. print("logr\_classifier",nltk.classify.accuracy(logr\_classifier,test))
93. sgd\_classifier=SklearnClassifier(SGDClassifier())
94. sgd\_classifier.train(train)
95. print("sgd\_classifier",nltk.classify.accuracy(sgd\_classifier,test))
96. linearsvc\_classifier=SklearnClassifier(LinearSVC())
97. linearsvc\_classifier.train(train)
98. print("linearsvc\_classifier",nltk.classify.accuracy(linearsvc\_classifier, test))
99. svc\_classifier=SklearnClassifier(SVC())
100. svc\_classifier.train(train)
101. print("svc\_classifier",nltk.classify.accuracy(svc\_classifier,test))
102. voteclass = voteclassifier(classifier,
103. mnb\_classifier,
104. bnb\_classifier,
105. linearsvc\_classifier,
106. sgd\_classifier,
107. svc\_classifier,
108. logr\_classifier)
109. print("voteclass",nltk.classify.accuracy(voteclass,test))
110. print("classification : ",voteclass.classify(test[0][0]),"confidence :", voteclass.confidence(test[0][0]))

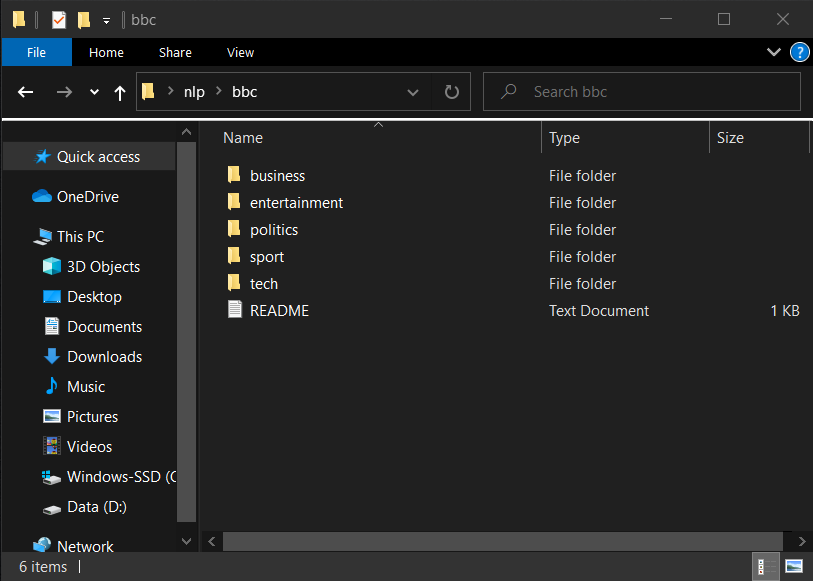
**Main.py**

1. import nltk
2. from nltk.tokenize import sent\_tokenize
3. import pickle
4. from textblob import TextBlob
5. l1 = []
6. business = {}
7. for i in range(1, 417):
8. if i<10:
9. k="0"+"0"+str(i)
10. elif i<100:
11. k="0"+str(i)
12. else:
13. k=str(i)
14. l1.append(k)
15. l = []
16. count = 0
17. count1 = 0
18. count2 = 0
19. score = 0
20. for i in l1:
21. with open("bbc/politics/"+str(i)+".txt") as file:
22. for read in file:
23. a=sent\_tokenize(read)
24. l.append(a)
25. for i in a:
26. analysis = TextBlob(i)
27. print(i, analysis.sentiment.polarity)
28. score = score+analysis.sentiment.polarity
29. print(score)
30. l = [j.replace("\n", "") for i in l for j in i]
31. line = ""
32. for i in l:
33. line = line+" "+str(i)
34. if score > 0:
35. count = count+1
36. business[line] = 1
37. elif score < 0:
38. count1 = count1+1
39. business[line] = -1
40. else:
41. count2 = count2+1
42. business[line] = 0
43. score = 0
44. line = ""
45. l = []
46. print(count, count1, count2)
47. with open("politic.pickle", 'wb') as dic:
48. pickle.dump(business, dic, protocol=pickle.HIGHEST\_PROTOCOL)

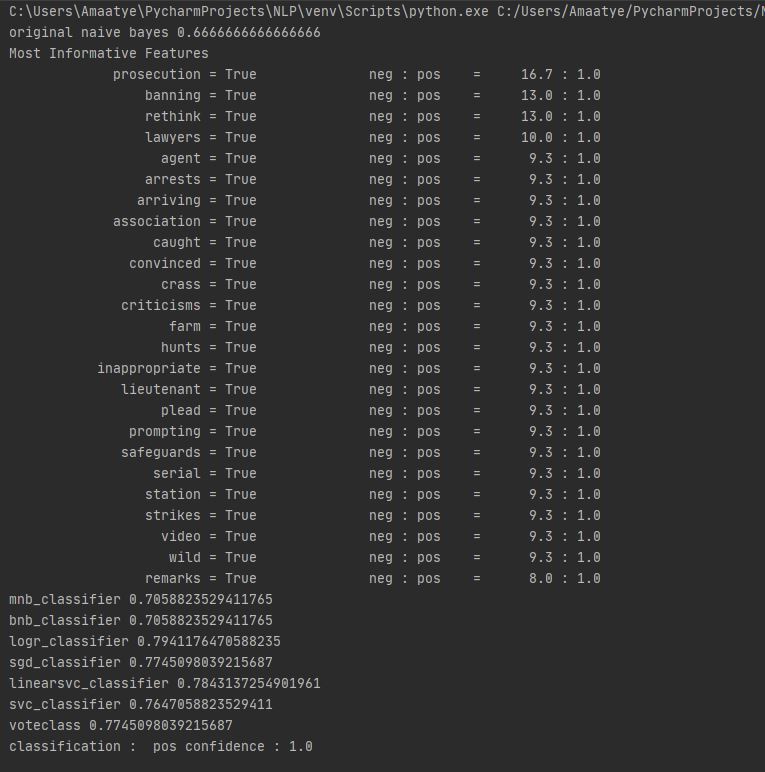
**Sentiment.py**

1. import nltk
2. from nltk.tokenize import sent\_tokenize
3. from textblob import TextBlob
4. from train import find\_feat, voteclass
5. l = []
6. score = 0
7. with open("new.txt") as file:
8. for read in file:
9. a = sent\_tokenize(read)
10. l.append(a)
11. for i in a:
12. analysis = TextBlob(i)
13. print(i, analysis.sentiment.polarity)
14. score = score+analysis.sentiment.polarity
15. print(score)
16. print("\n")
17. l = [j.replace("\n", "") for i in l for j in i]
18. line = ""
19. for i in l:
20. line = line + " " + str(i)
21. anal = TextBlob(line)
22. feat = find\_feat(line)
23. print("classification : ", voteclass.classify(feat), "confidence :", voteclass.confidence(feat))
24. score = 0
25. line = ""
26. l = []

**DATASETS USED**

****

**OUTPUT**



**LITERATURE SURVEY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AUTHOR** | **METHOD** | **PURPOSE** | **ADVANTAGES** | **DISADVANTAGES** |
| Xing Fang and Justin Zhan | sentiment polarity categorisation, by considering a dataset containing over 5.1 million product reviews | product reviews from amazon.com with the products belonging to four categories: beauty, books, electronics and home | The previous papers in this area suggested to remove all the objective content in order to conduct sentiment analysis but here, the subjective content is instead extracted for The performance of this approach is estimated by considering the average score. Therefore future work would be benefited if these limitations considered and thereby the accuracy and performance can future analysis | The performance of this approach is estimated by considering the average score. Therefore future work would be benefited if these limitations considered and thereby the accuracy and performance can future analysis be improved. |
| Linlin You and Bige Tuncer | CGSA:a Crowd-calibrated Geo-Sentiment Analysis mechanism) start the sentiment analysis process based on the design of CTS, and SSF | perform three analyses, namely sentiment, clustering and time series analysis on geo-tagged social network messages, and = collect crowd-labelled data based on a crowdsource d calibration service to gradually improve the classification accuracy | SSF has the best accuracy in training sentiment classifiers, and the performance of the calibrated classifier increases gradually and significantly from74.71%to 80.05% in three calibration cycles | Hundreds of thousands of users express opinions about companies, policies and announcements. However not all messages are relevant to the happenings in the market. Maintaining relevancy of the data to the current market scenario is a difficult task. Noise is another negative when it comes to analyzing messages online |
| Yang Peng, Melody Moh, and Teng-Sheng Moh | AD es are adverse drug events . Even though the pharmaceutical companies perform many drug-related tests before hand, when a drug is released into markets ADEs will be unidentified | data of four months on Twitter is collected so, as to capture the maximum number of ADEs | The drug classification is done for the cleaned data and the user opinion data is collected from which the ADEs are extracted. The captured tweets are stored in HIVE. | Combining different languages while tweeting, adding images and stickers in between the tweets and analysis of those tweets would be difficult |
| Arjit Chatterjee and Dr. William Perrizo | discuss the effect investors’ bias has on the volatility of stocks in the market | sentiment analysis was done on tweets of the potential investors and also why they used Microsoft Azure over other sentiment analyser tools. | Gets the tweets that are related to business management and tweets related to investors and investments | Twitter is one of the largest social media platforms with over 280 million active users with almost 500 million tweets created every day, training the program would be a difficult task |
| Tholana Venkata Satya Sai Abhishikth | A top-down approach is used to make sure a stock is not overrated or underrated by the investors | The approach is based on two broad behavioural finance assumptions-sentiment and limits to arbitrage | Through sentiment analysis, a particular user can understand the social sentiment score of aticker symbol based on the discussions of key investors and make an informed decision about which stock to invest | Because Microsoft Azure gave better results when compared too the r analysis tools, the popularity of this method is low. |
| Li Zhang, Liang Zhang, Keli Xhao and Qi Liu | DSA –Degree of Social Attention, which has been introduced by the previous works to capture stock price shocks. The method involves identifying the price shock as negative, near-zero or positive | Chinese Stock Market–SZSE and SSE are considered along with the social media activities inweibo.comin order to extend recent studies on financial activities in social media and their impact on the stock market | The efficiency of this work is proved by evaluating the F-measure without DSA and with DSA to show the improvement. The following are the classifiers used: Naïve Bayes, Decision Tree, Radom Forest, Logistic and LibSVM. Out of these classifiers, the Radom Foresthas the best performance and SVM the worst. | The proximity of the results may not be accurate while compared to other normal and manual methods , thus people with large number of data sets prefer the traditional method |

**CONCLUSION**

There are many directions in sentiment analysis that can be explored. This paper explored sentiment analysis of news and blogs using a dataset from BBC comprising of new articles of a year. It was observed that categories of business and sports had more positive articles, whereas entertainment and tech had a majority of negative articles. Future work in this regard will be based on sentiment analysis of news using various machine learning approaches with the development of an online application from where users can read news of their interests. Also, based on sentiment analysis methods, readers can customise their newsfeed

**REFERENCES**

[1]M. Karlsson, The immediacy of online news, the visibility of journalistic processes and a restructuring of journalistic authority. Journalism, 12(3), 279-295, 2011.

[2]A. Kohut, C. Doherty, M. Dimock and S. Keeter, Americans spending more time following the news. Pew Research Center,2010.

[3]J. Reis, P. Olmo, F. Benevenuto, H. Kwak, R. Prates, and J. An, Breakingthe news: first impressions matter on online news. In ICWSM ’15, 2015.

[4]N. Godbole, M. Srinivasaiah, and S. Sekine, Large-scale sentiment analysis for news and blogs. In International Conference on Weblogs and Social Media, Denver, CO, 2007.

[5]J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, “Towards buildinga social emotion detection system for online news,” Future Generation Computer Systems, vol. 37, pp. 438 448,2014.

[6]M. U. Islam, F. B. Ashraf, A. I. Abir and M. A. Mottalib, "Polarity detection of onlinenewsarticlesbasedonsentencestructureanddynamicdictionary,"201720thInternationalConferenceofComputerandInformationTechnology2017.8281777(ICCIT),Dhaka,2017,pp.1-5.doi:10.1109/ICCITECHN.

[7]V. Kharde, and P.Sonawane, “Sentiment analysis of twitter data: a survey of techniques,” International Journal of Computer Applications, vol. 139, no. 11, pp. 5–15,2016.

[8]B. Meyer, M. Bikdash, and X. Dai, “Fine-grained financial news sentiment analysis,” SoutheastCon 2017, 2017.

[9]V. S. Shirsat, R. S. Jagdale and S. N. Deshmukh, "Document Level Sentiment Analysis from News Articles," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, 2017, pp.1-4.

[10]A. Agarwal, V. Sharma, G. Sikka and R. Dhir, "Opinion mining of news headlines using SentiWordNet," 2016 Symposium on ColossalDataAnalysis and Networking (CDAN), Indore, 2016,pp.1-5.doi:10.1109/CDAN.2016.7570949

[11]J. Lei, Y. Rao, Q. Li, X. Quan, and L. Wenyin, “Towards building a social emotion detection system for online news,” Future Generation Computer Systems, vol. 37, pp. 438–448,2014.

[12]Musto, C., Semeraro, G., and Polignano, M, A comparison of lexicon-based approaches for sentiment analysis of microblog posts. Information Filtering and Retrieval, 59. 2014.

[13]A. Dandrea, F. Ferri, P. Grifoni, and T. Guzzo, “Approaches, tools andapplications for sentiment analysis implementation,” International Journal of Computer Applications, vol. 125, no. 3, pp. 26–33, 2015

[14]M. Devika, C. Sunitha, and A. Ganesh, “Sentiment analysis: a comparative study on different approaches,” Procedia Computer Science, vol. 87, pp. 44–49,2016.

[15]E.Haddi,X.Liu,andY.Shi,“TheRoleofTextPre-processingin Sentiment Analysis,” Procedia Computer Science, vol. 17, pp. 26–32, 2013.

[16]K.GhagandK.Shah,“SentiTFIDF–Sentimentclassificationusing relativetermfrequencyinversedocumentfrequency,”InternationalJournal of Advanced Computer Science and Applications, vol. 5, no. 2,2014.

[17]G.A.Miller,“WordNet:alexicaldatabaseforenglish,” Communications of the ACM, vol. 38, no. 11, pp. 39–41, Jan.1995.

[18]Z. Madhoushi, A. R. Hamdan, and S. Zainudin, “Sentiment analysis techniques in recent works,” 2015 Science and Information Conference (SAI), 2015.

[19] D.M.E.D.M. Hussein, “A survey on sentiment analysis challenges,” Journal of King Saud University -Engineering Sciences, vol. 30, no. 4, pp. 330–338,2018