Sentiment indicator for e-mails

K.D.Chandima

169304M

Faculty of Information Technology
University of Moratuwa
2019

Sentiment indicator for e-mails

K.D.Chandima

169304M

Supervisor: Dr. Lochandaka Ranathunga

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Honours Degree of Master Bachelor of Science in Information Technology.

2019

Declaration

I hereby declare that this project report entitled "Sentiment indicator for e-mails" contains my own work and has not been submitted and will not be submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student: K. D. Chandima

Signature of Student

Date: 25/04/2019

Supervised By

Dr. Lochandaka Ranathunga

Department of Information Technology
25 APR 2019
University of Moratuwa

Dedication

I would like to dedicate my project, "Sentiment indicator for e-mails";

To my project supervisor Dr. Lochandaka Ranathunga,

To my Father and sister who have given support for proof reading project documents,

To the Instructors of Faculty of Information Technology at University of Moratuwa who have supported me to find requirements for the project.

Acknowledgments

I wish to express my sincere thanks to my supervisor Dr. Lochandaka Ranathunga for guiding me to have a successful outcome in this project work.

I would like to acknowledge and extend my heartfelt gratitude to the following people who have made the completion of this project possible –

My father and sister for the help given to proof read all the project documents by spending their valuable time,

Instructors of Faculty of Information Technology at University of Moratuwa who have supported me to find resources and requirements for the project.

Academic, non-academic staff members who helped me throughout this project,

Batch mates who motivated in doing the project, and friends.

Abstract

People receive large amount of e-mails from friends, relatives, companies, institutions and known and unknown people. It is not possible the user to read all the received e-mails during a busy day. Normally, people avoid reading most of the received e-mails by giving priority to thc important e-mails. E-mails may contain happy news or sad news or defamatory contents or obscene contents. If there is a way for a user to get an idea of what kind of news would be there in just before he or she opens e-mails, then he or she can choose which e-mails should be opened first and find out which e-mails would make him happy. If the user can find out any obscene contents in an e-mail just before he opens it, then he or she can avoid the embarrassment of opening the email in front of a stranger and then the user will be able to delete it without simply opening it. A proper e-mail categorization is required and this attempt is to categorize e-mails according to the sentiment and the ingredient of the e-mail content, by indicating emoticons on the subject of e-mails. Text Preprocessing, Feature Extraction, Sentiment Classification methods are used to identify sentiments and proper emoticons are used to indicate sentiments on the subjects of e-mails. Machine learning and lexicon based approaches are used to predict the sentiment of emails. A better accuracy level is expected from machine learning in the process of sentiment extraction.

Table of Contents

Declaration i

Dedication ii

Acknowledgments iii

Abstract iv

Chapter 1 1

1.1 Introduction 1

1.2 Background 1

1.3 Aim and Specific Objectives 2

Chapter 2 3

Literature Review 3

2.1 Categorize e-mails by filtering Spam e-mails 3

2.2 E-mail classification 4

2.3 Categorize e-mails from Sentiment Analysis 4

2.4 Using Natural-Language-Processing (NLP) 5

2.5 Lexicon based sentiment analysis 6

2.5 Applications of Sentiment Analysis 6

Chapter 3 7

Technology adapted 7

3.1 Text preprocessing 7

3.2 Feature Extraction 7

3.3 Sentiment Classification 7

3.4 Natural-Language-Processing (NLP) 7

3.5 Graphical User Interface Creation 8

3.6 Reading E-mails from Gmail account 8

3.7 Lemmatization 8

3.8 Implementation of Neural Networks 8

3.9 Tools, Technologies and Libraries 9

3.10 Dataset 9

Chapter 4 10

Methodology 10

4.1 Statement of Research Problem 10

4.2 Expected outcome/alternative approaches 10

4.3 Reading E-mails 11

4.4 Text Preprocessing 11

4.4.1 Data Cleaning 12

4.4.2 Tokenization 13

4.4.3 Idiom handling 13

4.4.4 Negation handling 14

4.4.5 Part of Speech Tagging 14

4.4.6 Lemmatization 14

4.4.7 Stop word removal 14

4.5 Sentiment analysis using Lexicon approach 14

4.6 Machine Learning/ Deep Learning approach 15

4.6.1 Features selection 15

4.6.2 Neural Network 16

4.6.3 Training and testing datasets 16

4.7 Obscene image detection 16

Chapter 5 18

Analysis and Design 18

5.1 Text preprocessing 19

5.2 Feature Extraction 19

5.3 Sentiment Classification 19

5.4 System training and evaluation 19

5.5 Graphical User Interface (GUI) design 20

5.6 Neural Network 21

Chapter 6 22

Implementation 22

6.1 Reading emails from Gmail accounts 22

6.2 Creation of GUI 23

6.3 Data Preprocessing 24

6.3.1 Remove previously forwarded information from email body. 24

6.3.2 Removing images/links from the e-mail body 25

6.3.3 Remove punctuations 25

6.4 Detect obscene images from the body part of the image 26

6.5 Remove extra white spaces from email body 26

6.6 Tokenization 27

6.7 Lemmatization 27

6.8 Repetition handling 28

6.9 Idiom handling 28

Chapter 7 29

Evaluation and Results 29

7.1 Remove forwarded information method 29

7.2 Tokenize sentences method 30

7.3 Punctuation removing method 30

7.4 Repetition removal method 31

7.5 Sentiment extraction from Lexicon based approach 32

7.6 Sentiment extraction from Machine Learning Approach 33

7.7 Classifier Model 34

7.8 Accuracy 34

Chapter 8 35

Discussion 35

8.1 Discussion 35

8.2 Conclusion 35

8.3 Future works 36

References 38

List of Figures

Figure 4 1: emoticons 10
Figure 4 2: Previously forwarded details on emails 12

Figure 5 1: System Diagram 18
Figure 5 2: Graphical user interface with output 20
Figure 5 3: Neural Network Design 21

Figure 6 1: Source code to access Gmail API 22
Figure 6 2: Graphical User Interface implementation 23
Figure 6 3: Main emoticons 23
Figure 6 4: Forwarded e-mail 1 24
Figure 6 5: Forwarded e-mail 2 24
Figure 6 6: Source code for detect previously forwarded information 24
Figure 6 7: Source code for remove URLs from email body 25
Figure 6 8: Detecting obscene images from image URL 26
Figure 6 9: Source code for remove extra white spaces 26
Figure 6 10: Source code for tokenizing text into sentences and words 27
Figure 6 11: Lemmatization based on POS tags 27

Figure 7 1: E-mail body with previously forwarded information 29

Figure 7 2: After removing forwarded information 29
Figure 7 3: Tokenized sentences with line numbers 30
Figure 7 4: Punctuation, Symbols, Numbers removed output 30
Figure 7 5: Confusion Matrix for Machine Learning Approach 34

Chapter 1

1.1 Introduction

People receive large amount of e-mails from friends, relatives, companies, institutions and known and unknown people. It is not possible the user to read all the received e-mails during a busy day. Normally, people avoid reading most of the received e-mails by giving priority to the important e-mails. E-mails may contain happy news or sad news or defamatory contents or obscene contents. If there is a way for a user to get an idea of what kind of news would be there in just before he or she opens e-mails, then he or she can choose which e-mails should be opened first and find out which e-mails would make him happy. If the user can find out any obscene contents in an e-mail just before he opens it, then he or she can avoid the embarrassment of opening the email in front of a stranger and then the user will be able to delete it without simply opening it. The body part of the e-mail may contain defamatory content which might make the reader angry. A proper e-mail categorization is required and this attempt is to categorize e-mails according to the sentiment and the ingredient of the e-mail content, in-order to indicate emoticons on the subject of e-mails.

A methodology of indicating the sentiment or specific content in the e-mail will be a solution to this problem, The attempt of this research is to use Natural-Language-Processing (NLP) techniques to take out meaningful information from the body part of the e-mail and use Machine Learning Algorithms to ascertain the meaning of the text [1].

A proper e-mail categorization is required and this attempt is to categorize e-mails according to the sentiment and the ingredient of the e-mail content, by indicating emoticons on the subject of e-mails.

1.2 Background

E-mail services are some of the popular and fast communication methods evolved in internet. E-mail service providers are using client server architecture to provide e-mail services from e-mail server to e-mail clients. This Client-Server architecture enables high availability of email services. In order to communicate with one another, users need e-mail client programs. E-mails sent by the user and E-mails received by the user are synchronized among multiple computing and mobile devices which have e-mail client applications. These e-mail client applications display the subject of the e-mail to the user, so the user can have some idea about the e-mail by reading the subject of the e-mail. However, the body of the email contains broader information.

Natural-Language-Processing (NLP) techniques are used to take out purposive information by reading the body part of e-mails. Machine Learning Algorithms are used to ascertain the exact meaning of the text. These processes involve analyzing subjective and objective sentences and extract subject, adjectives, verbs, adverbs, nouns, pronouns prepositions, conjunctions. Nouns and verbs are specially considered to show that the email content has obscene content.

1.3 Aim and Specific Objectives

Aim: The aim is to perform sentiment analysis of emails and inform the user the sentiment and if the email has obscene content.

Objectives:

- To Extract sentiment by reading e-mail body using appropriate NLP and Machine Learning Technologies
- To develop an e-mail content summarization methodology to extract features.
- To classify e-mails by sentiments, reading e-mail body using appropriate NLP and Machine Learning Technologies.
- To train the system with pre-categorized e-mails and evaluate pre-categorized e-mails from the results of the system.
- To indicate sentiment and the ingredient of the e-mail by using "Emoticons" as icons on e-mail subject.
- To perform lexicon based sentiment analysis to compare accuracy.

Chapter 2

Literature Review

2.1 Categorize e-mails by filtering Spam e-mails

Anyone can send e-mails without any prior approval of the recipient and without authentication, this matter led the E-mail spam to become one of the growing problem in e-mail services [2]. E-mail spams are malware carriers and this make spams more dangerous than just filling inbox section of unnecessary e-mails [2]. Spam E-mails are used for information theft by attackers [3]. Spammers use spam domains to send malwares to the people who visit spam domains which provide false services and products [4]. An E-mail categorization mechanism would help the user to avoid spam e-mails.

Once the text of the body section has taken as the data set to classify the emails, first data cleaning should be performed as preprocessing process.

Researches which carried out to filter spam e-mails in the past, published various spam e-mail detection techniques. E.g. negative selection algorithm (NSA) with Differential Evolution(DE) [5], negative selection algorithm (NSA) combined with particle swarm optimization (PSO) [6], and search strategy subset of binary particle swarm optimization with mutation operator (MBPSO) [7].

To improve the performance of existing spam filtering techniques some of the data mining methods involved. For e-mail spam classification, "Decision Tree (DT) is used to handle nominal and numerical attributes, train data with missing attribute values and to increase efficiency of computing" [8]. Spam filtering can be performed using spam classification algorithms. LingerIG is an email classification system implemented in 2003 [9]. The context-based email classification model is another email classification method which classify emails and save them into several folders [10].

Spam messages can be identified by analyzing the body of e-mails using non-machine-learning-based spam filtering e.g. blacklisting/white listing or heuristic rule. Header-message-based spam filters have high accuracy compare to the body-message-based spam filter [11].

2.2 E-mail classification

There are mainly four approaches are there for e-mail classification "Traditional approach, Neural-Network approach, Graph-mining approach, Ontology-based approach" [10].

Traditional approach is related with Naive Bayes algorithm, which is a classifier constructing technique in the field of textual data analysis. Ontology-based approach is a way of summarization of text used to find most informative sentences. This sentence extraction maps to nodes of a hierarchical ontology [12]. Graph-mining approach convert e-mail content into graphs; Graph mining algorithms and data mining algorithms are used to discover patterns from those graphs [13]. Neural Network in text summarization: neural networks are trained in-order to learn and understand the sentences of e-mails.

Above approaches include several phases to categorize e-mails.

1. Preprocessing of e-mails - This involves Part_Of_Speech Tagging (POS Tagging) which is classifying words into their parts of speech [10].
2. Feature extraction - involves finding Sign-off words, Greeting words and keywords. Then transform e-mails to graphs and generate template graphs [10].
3. E-mail Classification – in this phase perform template ranking, matching graph to template graphs and place e-mails in relevant folder [10].

2.3 Categorize e-mails from Sentiment Analysis

Sentiment analysis is an area of research that widely used to monitor online content and social media in-order to determine opinion and the emotional tone of sentences. Natural-Language-Processing (NLP), artificial neural networks and text analysis methods are evolved as supporting fields for sentiment analysis. Emotions are classified mainly into tree types, Positive, negative and neutral [14]. The emoticons that can be relevant for these emotions are "happy, sad, pleasant, fear, satisfied and frustrated" [14].

There are two methods used in the field of Sentiment Analysis and Opinion Mining

1. Lexicon Based Approach
2. Machine learning Approach

Lexicon Based Approach does not require the data set to be trained as a data preprocessing process. In this approach mainly indicate negative and positive sentiment of words. Two different approaches can be denoted under "Lexicon Based Approach"; Dictionary Based Approach and Corpus Based Approach. In Dictionary Based Approach, lexicon like 'WordNet' is used to search synonym and antonym of words [15]. The Corpus Based Approach can be used to solve the problems of Dictionary Based Approach, but it require huge corpus. So the Corpus Based Approach is not efficient as dictionary based approach and creating a huge corpus is difficult [15]

2.4 Using Natural-Language-Processing (NLP)

These processes involve analyzing subjective and objective sentences and extract subject, adjectives, verbs, adverbs, nouns, pronouns prepositions, conjunctions. Nouns and verbs are specially consider to show that the email content has technical contents or obscene content.

Separation of subjective sentences and objective sentences are performed by giving more weight on subjective sentences while determining the sentiments [16]. There is a possibility to apply Sentimental Analysis in different levels such as document level, Sentence level and Phrase level [16].

2.5 Lexicon based sentiment analysis

Lexicons which contain words and the relevant sentiment can be used to identify the sentiment of sentences or documents. Lexicons are dictionaries and generally they contain positive and negative polarity of words. Saif M. Mohammad and Peter D. Turney created large, high quality word-emotion lexicon consist emotions of joy, sadness, anger, fear, trust, disgust, surprise, and anticipation [17]. This lexicon contains 14183 words with relevant emotion and this lexicon is called as "NRC Lexicon". There is another lexicon called "WordNet Affect Lexicon (WAL)" and it has few hundreds of words and six emotions (anger, disgust, fear, joy, sadness, surprise) associated with them. Lexicon based sentiment analysis express nearly same accuracy as machine learning based methods provide.

2.5 Applications of Sentiment Analysis

Several applications of Sentiment Analysis can be listed as follows.

The American travel website TripAdvisor is using sentiment analysis to analysis traveler's reviews and extract visitor attractions and popular restaurants [18]. Analysis of writing patterns for multi-class classification using SENTA depict usage of tool built to help users select out of a wide variety of features the ones that fit the most for their application, to run the classification, through an easy-to-use graphical user interface. [19] [20] [21]. Movie reviews mining is another application area sentiment classification by analyzing reviewers opinions and determining whether the opinions are positive or negative [22]. Multimodal sentiment analysis is another application area of opinion mining methodology for video and audio using Additive Learning Algorithm [23].

Chapter 3

Technology adapted

3.1 Text preprocessing

This process performs mainly using NLTK library and regular expressions. Sentence tokenizing, word tokenizing and “Stop-Word-Removal” were achieved via NLTK library. Common text preprocessing functions are available in Python NLTK library. Regular expressions were used to remove punctuations of the text. TextBlob which is a python library was used to correct spellings which is a part of text preprocessing.

3.2 Feature Extraction

Features were extracted from cleaned input data. Machine learning algorithms were used to generate a “classifier model” which is based on labels and “features set” [24]. Sentiment prediction was done extracting “features set” and feeding them to a classifier model.

3.3 Sentiment Classification

Sentiments were analyzed by lexicon based approach and “Multilayer Neural Networks”. The Recursive model of Neural Network has a hidden layer and uses bottom-up approach for sentiment analysis [25]. Based on the number of words of the sentence, a percentage of the sentiment type was calculated from a lexicon based approach.

3.4 Natural-Language-Processing (NLP)

Natural-Language-Processing (NLP) techniques and Machine Learning Algorithms were used to exact meanings of the text by reading the body part of e-mails.

3.5 Graphical User Interface Creation

Graphical user interfaces were created using Tkinter the Python's standard "GUI Programming Toolkit". Tkinter provides Object-Oriented techniques of creating GUIs.

3.6 Reading E-mails from Gmail account

Gmail account access was performed through Gmail API. Gmail API provides RESTful access to emails. It was possible to read and send messages, make attachments and manage Gmail settings via the Gmail API. Client configuration file called "credentials.json" was downloaded through Gmail API in-order to authenticate the Gmail user. Then a python application could read and send emails through this json file. The API user guide page provided some sample codes to show how emails are read using this "credentials.json" file and this source code was downloaded from https://developers.google.com/gmail/api/quickstart/python.

3.7 Lemmatization

The process of lemmatization was performed using the library "WordNetLemmatizer". Lemmatization was performed considering parts-of-speech tags available in WordNet corpus.

3.8 Implementation of Neural Networks

The neural network used to classify sentiments was created by using the library called numpy. Random value generation, calculation of exponentials, perform matrix manipulation and creation of numpy arrays were some functions used in the creation of the neural network.

3.9 Tools, Technologies and Libraries

- NLTK python library was used to perform tokenization, stemming, part of speech tagging, parsing, classification and semantic reasoning functionalities
- **Tools and Libraries:** Packages of Anaconda Distribution, NLTK, SCIKIT-learn
- **IDE:** Pycharm, Visual Studio
- **Programming languages:** Python, C#, JSON

3.10 Dataset

- Main dataset – Enron email corpus, has 1million emails of 158 employees of Enron Corporation [26].
- Angry Letters [27].
- Hate Mails – Haters email contains emails from some people opposed to vaccination. [28].
- https://www.whitman.edu/VSA/letters/
- https://americanbridgepac.org/jeb-bushs-gubernatorial-email-archive/
- Love letters
- https://sweetlovemessages.com/short-love-letters-him-her
- https://www.writeexpress.com/ILoveUad.html
- https://www.writeexpress.com/apolog06.html

Chapter 4

Methodology

4.1 Statement of Research Problem

The body part of the e-mail may contain a happy news or a sad news or a defamatory content which might make the reader distressed. Sometimes the e-mails may contain obscene quotes and pictures. If there is a way for a user to get an idea of the kind of news that would be there prior to opening e-mails, then he or she can choose which e-mails should be opened first.

4.2 Expected outcome/alternative approaches

A software system was created to extract the sentiments by reading body part of e-mails and finding out whether the e-mails contain a happy news, sad news, defamatory content or an obscene content. The software system is capable of displaying an icon on each e-mail in order to indicate the sentiment or the specific content before the user opens it. Emoticons on e-mail subject are used to indicate happy news, sad news, defamatory content, and obscene content.

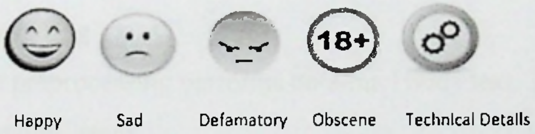


Figure 4.1: emoticons

These sentiments (Figure 4.1) are extracted using two different approaches. They are Lexicon based approach and Machine learning approach.

Technical details contain a detailed description about the sentiment. If the defamatory details are there in the e-mail body, these details may present abusive language towards someone directly or someone else. A sad news can be present in the first half of the e-mail and in the second half, defamatory details can be present. The details of "How the ingredients of the e-mail have been arranged" are indicated under "Technical Details".

Obscene content may appear in the body part of the e-mail or in the attachment as images and videos. These details are listed separately under Technical Details Icon.

The initial input e-mail data set is a list of e-mails received to some of my personal e-mail client. System Training and Evaluation was done by using online corpus data set

Taken from https://www.cs.cmu.edu/~./enron/.

4.3 Reading E-mails

The software system starts with reading e-mails from a Gmail account. The Gmail account is a specially created Gmail account for this research and the e-mail address is sendicator169304m@gmail.com .

Gmail API was used in order to access this Gmail account. Gmail API provided RESTful access to emails. It was possible to read and send messages via the Gmail API. Client configuration file called "credentials.json" was downloaded through Gmail API in-order to authenticate the Gmail user. Then the python application/ the system read emails through this json file as plain text.

4.4 Text Preprocessing

In this stage, text preprocessing performs on e-mail body text. Text preprocessing stage has several sub stages/ steps.

1. Data Cleaning

    1.1. Detect, save and remove http, https, ftp links/ tags and email addresses

    1.2. Save and delete previously forwarded information from e-mail body

    1.3. Remove extra white spaces and new line characters from e-mail body

    1.4. Spelling correction

    1.5. Handling of repetition of characters in each word

    1.6. Remove special characters from text.

2. Tokenization
3. Idiom handling
4. Negation handling
5. Part of Speech Tagging
6. Lemmatization
7. Stop word removal

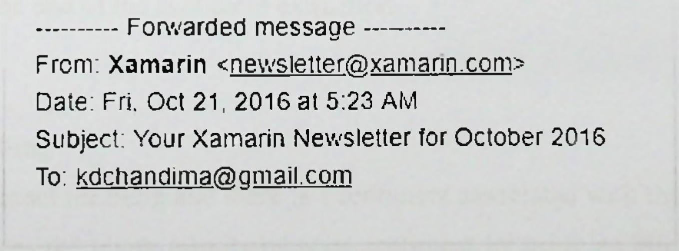
4.4.1 Data Cleaning

- **Detect, save and remove http, https, ftp links/ tags and email addresses**

  An email can contain obscene images and the system detects these obscene images by reading its http, https or ftp links/ tags. Once the image links were detected by the system, it reads the domains of each link to determine whether the image is obscene or not. These http, https, ftp, xml and email addresses in the e-mail body are not necessary to detect the sentiment of the email and thus, the system remove them. Those links are saved in the system before they are removed. Therefore, the saved image links can be used to regenerate the original email to user once the sentiment analysis process is completed.

- **Save and delete previously forwarded information from e-mail body**

  Body part of the email may contain previously forwarded information.



---------- Forwarded message ----------
From: **Xamarin** <newsletter@xamarin.com>
Date: Fri, Oct 21, 2016 at 5:23 AM
Subject: Your Xamarin Newsletter for October 2016
To: kdchandima@gmail.com

Figure 4 2: Previously forwarded details on emails

These previously forwarded information are irrelevant to detect the sentiments from email body. The system removes these information from the body part of the emails and saves them for later use. At the end of sentiment analysis process, the original email is displayed to the user. The system uses the saved previously forwarded information to regenerate the original email at the end of the sentiment analysis process.

- **Remove extra white spaces and new line characters from e-mail body**

  After removing xml tags and email addresses from email body, there will be extra white spaces and new lines on the email body. The system removes them to avoid errors which can be generated while extracting sentiments.

- **Spelling correction and repetition handling**

  Spelling mistake correction and the removal of repetition of consecutive letters in each word of email body was performed in this stage.

- **Remove special characters from text.**

  The system removes punctuations, alphanumeric characters, symbols and numbers from email body.

4.4.2 Tokenization

After data cleaning stage was completed, the system tokenizes the email body into sentences and then those sentences are further tokenized into words. Tokenized sentences are stored in a python list and the line number of each sentence is kept unchanged until the end of the sentiment extraction.

4.4.3 Idiom handling

An idiom has an exact meaning and there is a sentiment associated with this meaning. The system replaces the idiom into its relevant sentiment by using an idiom lexicon. The lexicon was created manually considering the meanings of idioms.

4.4.4 Negation handling

The word phrases like "ain't, aint, can't, cant, couldn't, didn't, doesn't, don't, dont, hasn't, haven't, never, no, not, nothing and won't" change the sentiment of a word to its opposite. Handling these word phrases is important to identify the sentiment of a sentence accurately. Hence the system replaces these word phrases to a single word "not".

4.4.5 Part of Speech Tagging

Nouns, adjectives, verbs and adverbs of a sentence are called features and the system feeds them to a neural network as inputs. Hence POS tagging is important to find these features.

4.4.6 Lemmatization

Reducing the inflectional forms of each word into a common base or root is necessary to have better accuracy in sentiment extraction. All the words of the lexicon which is used to extract sentiments in this project are present in their common base. The system compares email body text with the lexicon to detect the proper sentiment in the lexicon approach. The system performs lemmatization of email content according to their part of speech tag in order to have higher accuracy in sentiment extraction.

4.4.7 Stop word removal

Stop words are not necessary in the process of sentiment extraction. The system removes all the stop words at the end of the text preprocessing stage.

4.5 Sentiment analysis using Lexicon approach

The lexicon used in this project is a combination of two different lexicons. First lexicon is NRC Word-Emotion Association Lexicon (NRC Emotion Lexicon) [29] and second lexicon is WordNet Affect Lexicon [30]. The combined lexicon has more than 4000 words and 6 sentiments associated with them (anger, joy, sadness, obscene, negative and positive).

In the process of sentiment extraction, the system creates a bag of words from the combined lexicon. They are called happy-bag, sad-bag, defamatory-bag, obscene-bag, negative-bag and positive-bag. Then each tokenized word from each sentence is checked in these bags to detect the sentiment associated with them. Then a count is taken for each sentiment for each sentence. Then the two sentiments that have higher probabilities will be taken as the predicted labels for a particular sentence. The sentiment that has higher probability for the whole email body will be considered as the predicted sentiment for the email. This sentiment is indicated on the subject of the email using an emoticon. Even if an email has an obscene content in a low probability, the sentiment will be indicated on the subject of the email using an emoticon.

4.6 Machine Learning/ Deep Learning approach

In this approach, the system first selects the features and then extracts these features from each sentence of email body text. The features are then sent to an artificial neural network as inputs.

4.6.1 Features selection

When a sentence is processed, the words related to nouns, adjectives, verbs, adverbs and negation (not) are separated into categories using POS tagging. Each category may have a mixture of happy words, sad words, defamatory words and obscene words. This mixture is converted into a number in order to input into an artificial neural network. To give a number for each category, the bag of words have been used which are created using the combined lexicon. Word count in each bag of word can be denoted as follows,

Obscene-bag = 445

Happy-bag = 1023

Sad-bag = 1195

Defamatory-bag = 1476

It was assumed that one sentence will have a maximum of 20 words relevant to each sentiment for each category.

When giving numbers for every word, the summation of two different mixtures should not be equal.

Then the words of obscene-bag were numbered from 1 to 445

Happy-bag was numbered from 30,000 to 33,000,000

Sad-bag was numbered from 660,000,000 to 825,000,000,000

Defamatory-bag was numbered from 16,500,000,000,000 to 24,750,000,000,000,000

There is another category called negation and this category only contains the word 'not'.

The word 'not' was numbered as -1

The summation of the numbers of the mixture of each category becomes a feature in the Machine Learning Approach.

4.6.2 Neural Network

A neural network was created as a classification model and it has three layers (input layer, output layer and hidden layer). Input layer has five nodes and the five features (nouns, adjectives, verbs, adverbs and negation) are the inputs which feeds them. The neural network has four outputs (Happy, Sad, Defamatory and obscene) and the outputs are probability values.

4.6.3 Training and testing datasets

The training dataset contains 100 defamatory e-mails, 200 happy emails, 150 sad emails, 10 obscene emails, 43 sentiment neutral emails. Apart from this email dataset, labeled movie reviews were used to train the neural network.

The testing dataset contains 14 defamatory emails, 31 happy emails, 27 sad emails, 26 sentiment neutral emails, 83 emails of which the sentiment is difficult to detect directly.

4.7 Obscene image detection

Images on email body has a URL to the online source. In the **"http, https, ftp links detection stage"**, these URLs were saved within the software system. A URL contains a domain name and this domain was used to identify the source web site.

Eg:

<https://www.google.com/appserve/mkt/p/AJ-PF7yvhcl0DqNw_vOfWiwv6Glmy >

Source web site/ domain name - www.google.com

The system maintains a list of adult web site domains, which contain 109 adult web sites. The system checks the image domain name in this list and if the image domain is available in the list, it is marked as obscene. This information is indicated on the email subject to inform the user.

Chapter 5

Analysis and Design

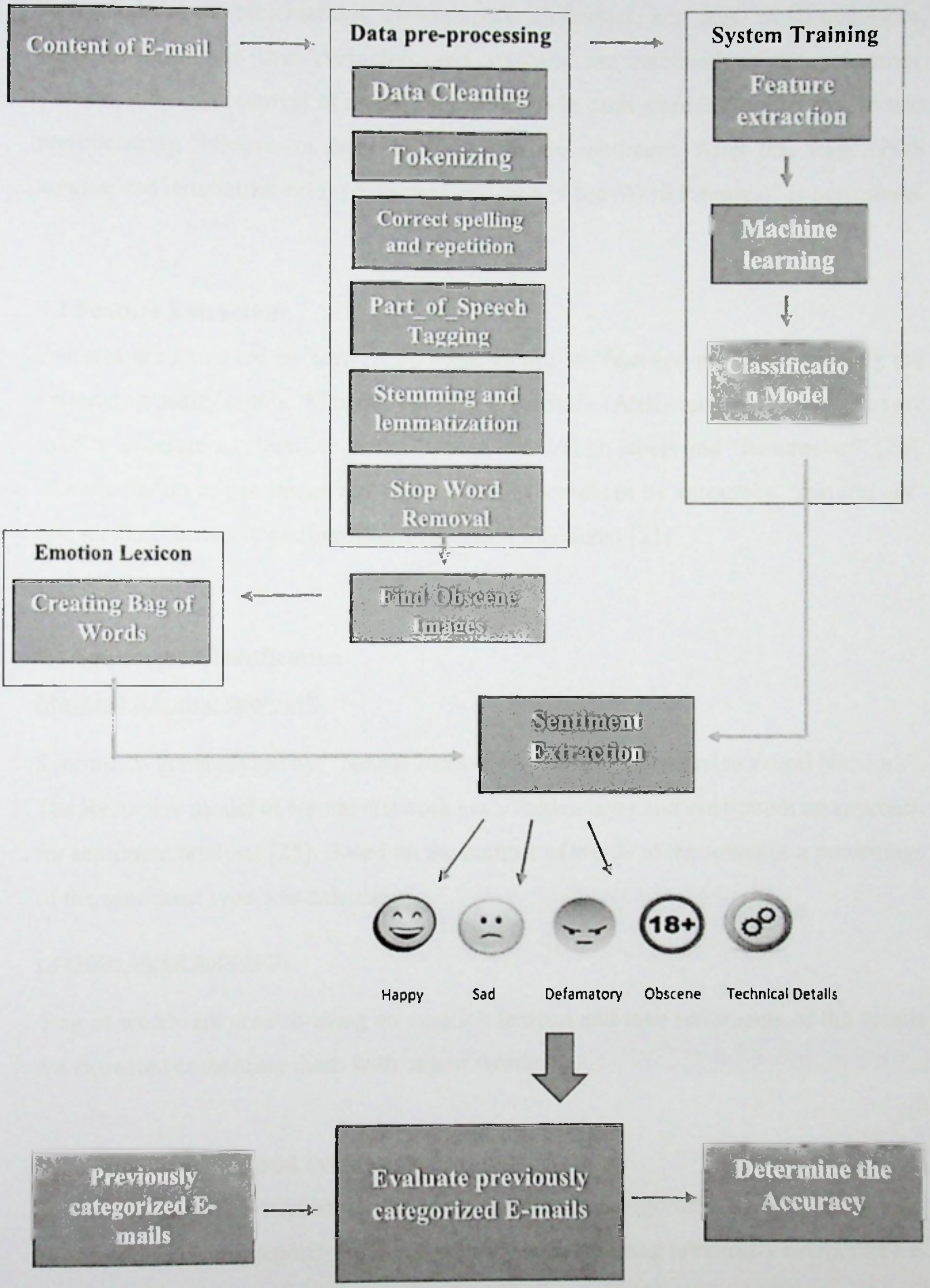


Figure 5 1: System Diagram

5.1 Text preprocessing

Data cleaning was done as the first step of the Text Preprocessing stage. Using "Regular Expressions", unnecessary information of the e-mail body was removed. Previously forwarded details, punctuations, alphanumeric characters, xml tags, email addresses, white spaces, new line characters and symbols are removed as data cleaning. Tokenization and removal of repetition characters in each word is the next step in text preprocessing. Idioms are then replaced with the sentiment. After this stage, POS tagging and lemmatization is performed and then "Stop-Word Removal" is performed.

5.2 Feature Extraction

Features are extracted by converting input values to "features set" while training the system to classify inputs. Machine learning algorithms (Artificial Neural Network) are used to generate a "classifier model" which is based on labels and "features set" [24]. The prediction of sentiment was done for input sentences by extracting "features set" and feeding them to classifier model (decision functions) [31].

5.3 Sentiment Classification

Machine learning approach

Sentiments are analyzed by, "Neural Network Systems" or "recursive neural Network". The Recursive model of Neural Network has a hidden layer and use bottom up approach for sentiment analysis [25]. Based on the number of words of the sentence a percentage of the sentiment type was calculated.

Lexicon based approach

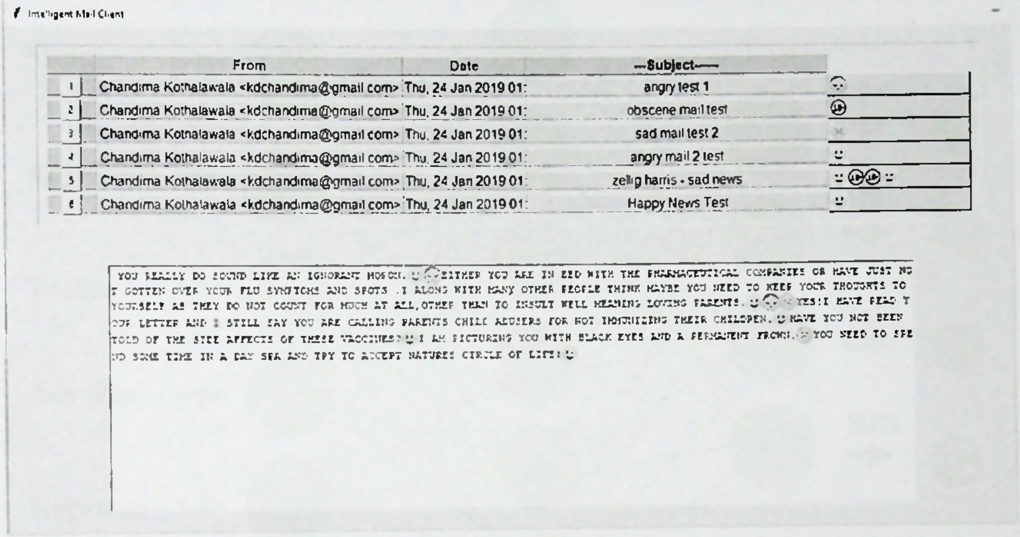
Bag of words are created using an emotion lexicon and then sentiments of the emails are extracted comparing them with bag of words.

5.4 System training and evaluation

System training was performed by using previously categorized e-mails taken from online corpus e-mail repository. The System was tested using previously categorized e-mails. Once the system has produced outputs from machine learning and lexicon based methods, then the accuracy of outputs of those two methods were determined by previously categorized e-mails.

5.5 Graphical User Interface (GUI) design

A graphical user interface was created in the client application to display the emails in user's email account. The sentiment of each email is indicated on the subject of the emails using an emoticon as shown in the figure 5.2.



	From	Date	—Subject——
1	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	angry test 1
2	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	obscene mail test
3	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	sad mail test 2
4	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	angry mail 2 test
5	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	zellig harris - sad news
6	Chandima Kothalawala <kdchandima@gmail com>	Thu, 24 Jan 2019 01:	Happy News Test

YOU REALLY DO SOUND LIKE AN IGNORANT MORON. EITHER YOU ARE IN BED WITH THE PHARMACEUTICAL COMPANIES OR HAVE JUST NOT GOTTEN OVER YOUR FLU SYMPTOMS AND SPOTS .I ALONG WITH MANY OTHER PEOPLE THINK MAYBE YOU NEED TO KEEP YOUR THOUGHTS TO YOURSELF AS THEY DO NOT COUNT FOR MUCH AT ALL,OTHER THAN TO INSULT WELL MEANING LOVING PARENTS. YES!I HAVE READ YOUR LETTER AND I STILL SAY YOU ARE CALLING PARENTS CHILD ABUSERS FOR NOT IMMUNIZING THEIR CHILDREN. HAVE YOU NOT BEEN TOLD OF THE SIDE AFFECTS OF THESE VACCINES? I AM PICTURING YOU WITH BLACK EYES AND A PERMANENT FROWN. YOU NEED TO SPEND SOME TIME IN A DAY SPA AND TRY TO ACCEPT NATURES CIRCLE OF LIFE!

Figure 5 2: Graphical user interface with output

5.6 Neural Network

The neural network used in this software system has three layers which are input layer, one hidden layer and output layer. Input layer has five nodes into which five features are fed. The five features are nouns, adjectives, verbs, adverbs and negation ('not'). The output layer has four output nodes to output probability values relevant to sentiments (defamatory, happy, sad, and obscene). The hidden layer consists of six nodes. Figure 5.3 shows the design of the neural network.

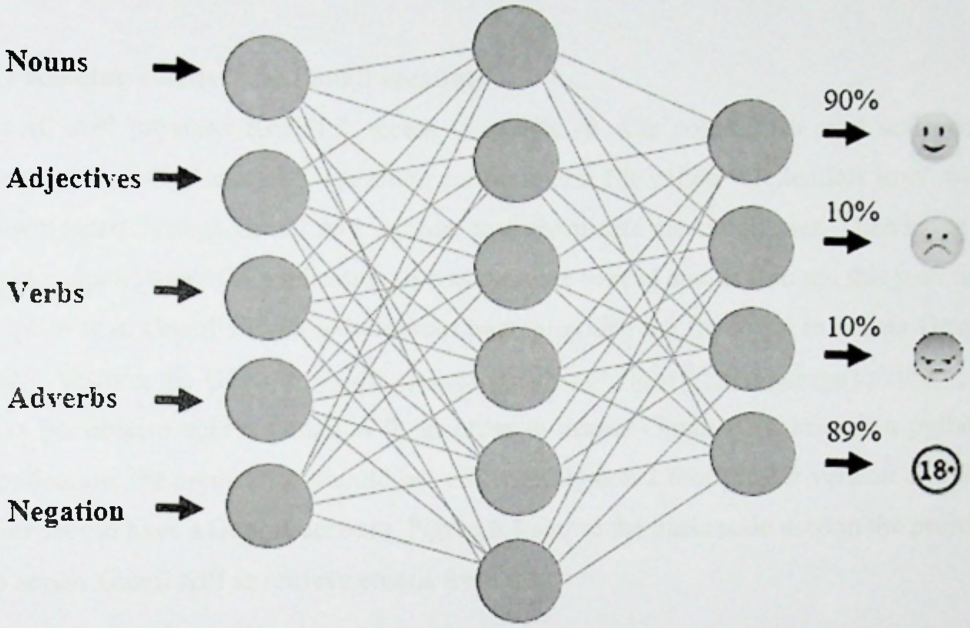


Figure 5.3: Neural Network Design

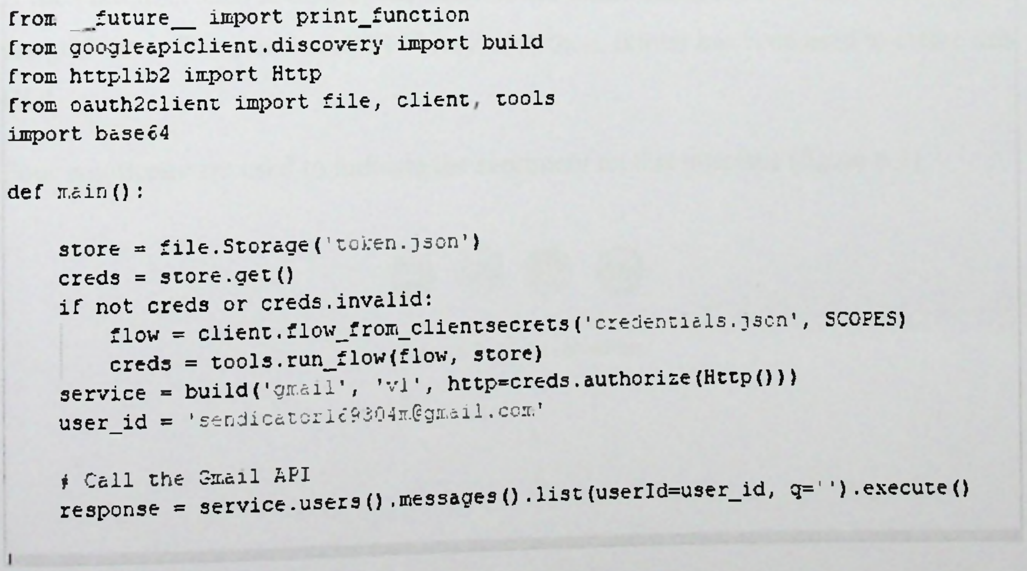
Chapter 6

Implementation

Python programming language was used for development of this project, and python provides a great support in natural language processing. Python has a massive number of various libraries that makes it easier to conduct NLP tasks in more efficient ways. Python provides the powerful NLTK library for natural language processing tasks.

6.1 Reading emails from Gmail accounts

Gmail API provides RESTful access to emails. It was possible to read and send messages via the Gmail API. A client configuration file called "credentials.json" was downloaded through Gmail API in-order to authenticate the Gmail user. Then using a python application/ C# application, it was possible to read emails through this json file in plain text. Gmail's Python Quickstart page provided the functions to access Gmail API. Visiting the URL https://developers.google.com/gmail/api/quickstart/python , it was possible to access Gmail API. In order to access Gmail API through a python application, the application should be written in Python 2.6 or greater version and the user should have a Google account. Figure 6.1 shows the basic code used in the project to access Gmail API to retrieve emails from inbox.

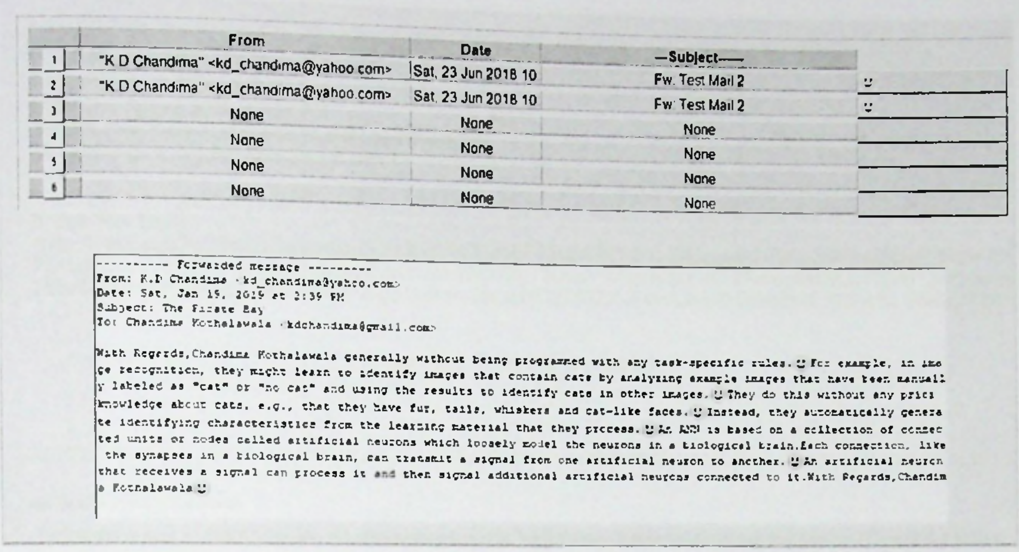


```
from __future__ import print_function
from googleapiclient.discovery import build
from httplib2 import Http
from oauth2client import file, client, tools
import base64

def main():

    store = file.Storage('token.json')
    creds = store.get()
    if not creds or creds.invalid:
        flow = client.flow_from_clientsecrets('credentials.json', SCOPES)
        creds = tools.run_flow(flow, store)
    service = build('gmail', 'v1', http=creds.authorize(Http()))
    user_id = 'sendicator169304m@gmail.com'

    # Call the Gmail API
    response = service.users().messages().list(userId=user_id, q='').execute()
```

Figure 6 1: Source code to access Gmail API

6.2 Creation of GUI

Figure 6.2 show the GUI created in the project to display outputs.



	From	Date	—Subject—	
1	"K D Chandima" <kd_chandima@yahoo.com>	Sat, 23 Jun 2018 10	Fw. Test Mail 2	☺
2	"K D Chandima" <kd_chandima@yahoo.com>	Sat, 23 Jun 2018 10	Fw. Test Mail 2	☺
3	None	None	None	
4	None	None	None	
5	None	None	None	
6	None	None	None	

```
---------- Forwarded message ---------
From: K.D Chandima <kd_chandima@yahoo.com>
Date: Sat, Jan 19, 2019 at 2:39 PM
Subject: The Pirate Bay
To: Chandima Kothalawala <kdchandima@gmail.com>

With Regards,Chandima Kothalawala generally without being programmed with any task-specific rules. ☺ For example, in ima
ge recognition, they might learn to identify images that contain cats by analyzing example images that have been manuall
y labeled as "cat" or "no cat" and using the results to identify cats in other images. ☺ They do this without any prior
knowledge about cats, e.g., that they have fur, tails, whiskers and cat-like faces. ☺ Instead, they automatically genera
te identifying characteristics from the learning material that they process. ☺ An ANN is based on a collection of connec
ted units or nodes called artificial neurons which loosely model the neurons in a biological brain.Each connection, like
 the synapses in a biological brain, can transmit a signal from one artificial neuron to another. ☺ An artificial neuron
that receives a signal can process it and then signal additional artificial neurons connected to it.With Regards,Chandim
a Kothalawala ☺
```

Figure 6 2: Graphical User Interface implementation

This user interface has two primary sections, the grid-view on the top and the text box located at the bottom of the GUI. The grid view displays the compact details of all the emails. It also displays the sender of each email, date the email was sent, subject of the email and the extracted sentiment from performing sentiment analysis. The text widget at the bottom is used to display the body of the email which the user has selected from the grid-view. The standard GUI library of python, tkinter has been used to create this GUI.

Four emoticons are used to indicate the sentiment on this interface (figure 6.3).

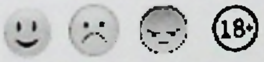


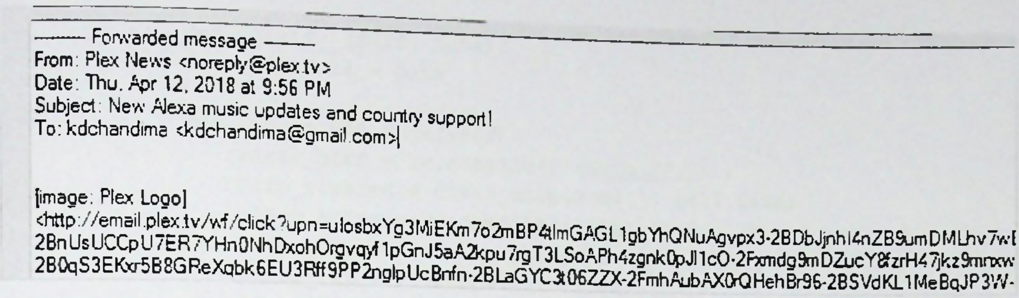
Figure 6 3: Main emoticons

6.3 Data Preprocessing

6.3.1 Remove previously forwarded information from email body.

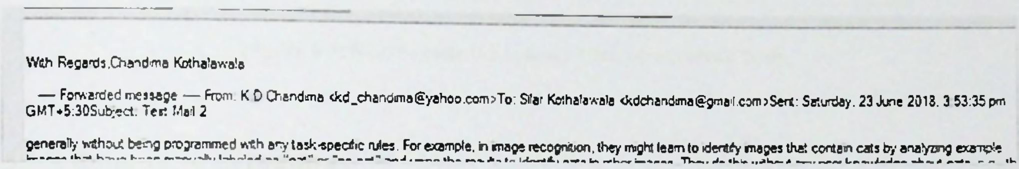
Previously forwarded information is unnecessary to identify sentiment of the e-mail.

Eg:



```
------- Forwarded message ---------
From: Plex News <noreply@plex.tv>
Date: Thu, Apr 12, 2018 at 9:56 PM
Subject: New Alexa music updates and country support!
To: kdchandima <kdchandima@gmail.com>


[image: Plex Logo]
<http://email.plex.tv/wf/click?upn=ulosbxYg3MiEKm7o2mBP4lmGAGL1gbYhQNuAgvpx3-2BDbJjnhl4nZB9umDMLhv7wI
2BnUsUCCpU7ER7YHn0NhDxohOrgvqyf1pGnJ5aA2kpu7rgT3LSoAPh4zgnk0pJl1cO-2Fxmdg9mDZucY8fzrH47jkz9mnxw
2B0qS3EKxr5B8GReXqbk6EU3Rff9PP2nglpUcBnfn-2BLaGYC3t06ZZX-2FmhAubAX0rQHehBr96-2BSVdKL1MeBqJP3W-
```

Figure 6 4: Forwarded e-mail 1



With Regards,Chandima Kothalawala

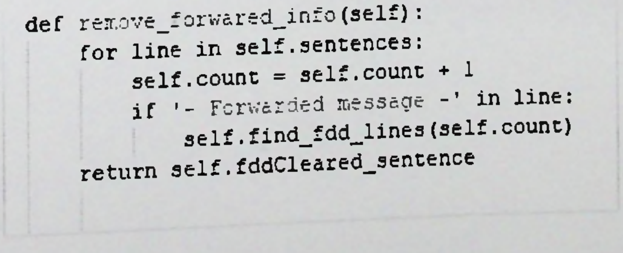
— Forwarded message — From: K D Chandima <kd_chandima@yahoo.com>To: Silar Kothalawala <kdchandima@gmail.com>Sent: Saturday, 23 June 2018, 3:53:35 pm GMT+5:30Subject: Test Mail 2

generally without being programmed with any task-specific rules. For example, in image recognition, they might learn to identify images that contain cats by analyzing example

Figure 6 5: Forwarded e-mail 2

This information is removed as the first step of data preprocessing and python's string matching techniques were used to do this task.

In order to remove this section, e-mail body is tokenized into text lines and text lines are identified which have the pattern "**- Forwarded message -**". Then the system checks the proceeding keywords "**From:**", "**Date:**", "**Subject:**", "**To:**" and if they are present, the details of these sections will be removed from the email body (see Figure 6.6).

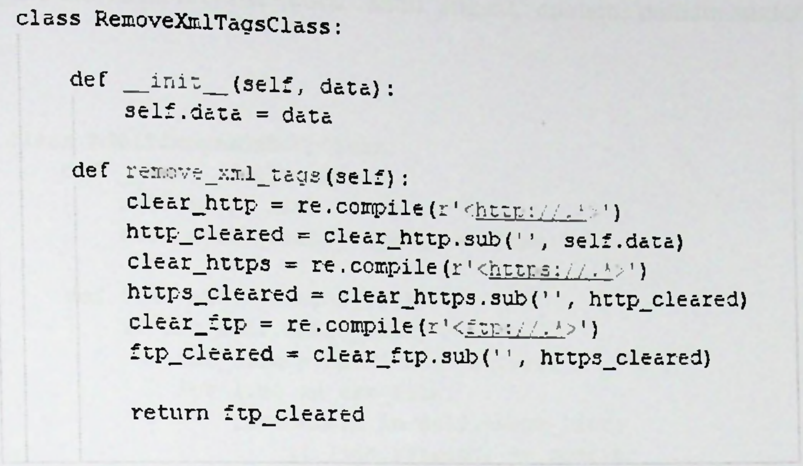


```
def remove_forwared_info(self):
    for line in self.sentences:
        self.count = self.count + 1
        if '- Forwarded message -' in line:
            self.find_fdd_lines(self.count)
    return self.fddCleared_sentence
```

Figure 6 6: Source code for detect previously forwarded information

6.3.2 Removing images/links from the e-mail body

Regular expressions were used to identify and remove different web URLs and xml tags. The figure 6.7 shows how the system removes URLs from email body.



```
class RemoveXmlTagsClass:

    def __init__(self, data):
        self.data = data

    def remove_xml_tags(self):
        clear_http = re.compile(r'<http://.*>')
        http_cleared = clear_http.sub('', self.data)
        clear_https = re.compile(r'<https://.*>')
        https_cleared = clear_https.sub('', http_cleared)
        clear_ftp = re.compile(r'<ftp://.*>')
        ftp_cleared = clear_ftp.sub('', https_cleared)

        return ftp_cleared
```

Figure 6 7: Source code for remove URLs from email body

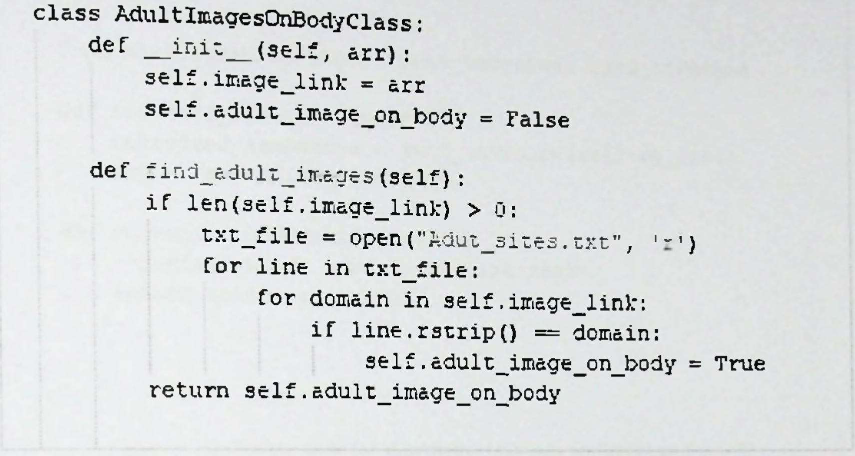
6.3.3 Remove punctuations

The punctuation removal was performed by considering all the symbols and numbers.

Punctuations = '''!#$%^&*()_+-=[]{};:'"|<>,.?/1234567890V'''

6.4 Detect obscene images from the body part of the image

Obscene images were identified by reading the domain of the url of images. These domain names were then crosschecked with a list of known obscene web site domain names (figure 6.8). The text file called "adult_site.txt" contains domain names of adult web sites.

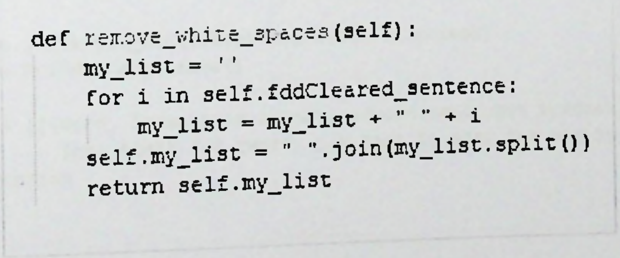


```
class AdultImagesOnBodyClass:
    def __init__(self, arr):
        self.image_link = arr
        self.adult_image_on_body = False

    def find_adult_images(self):
        if len(self.image_link) > 0:
            txt_file = open("Adut_sites.txt", 'r')
            for line in txt_file:
                for domain in self.image_link:
                    if line.rstrip() == domain:
                        self.adult_image_on_body = True
        return self.adult_image_on_body
```

Figure 6 8: Detecting obscene images from image URL

6.5 Remove extra white spaces from email body

Extra white spaces are created after removing images from the email body. To remove these extra white spaces the system uses join and split function in python language. The source code related to removing white spaces is shown in figure 6.9.

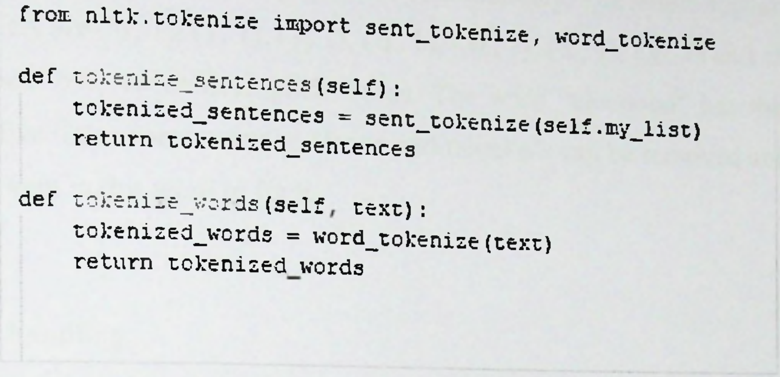


```
def remove_white_spaces(self):
    my_list = ''
    for i in self.fddCleared_sentence:
        my_list = my_list + " " + i
    self.my_list = " ".join(my_list.split())
    return self.my_list
```

Figure 6 9: Source code for remove extra white spaces

6.6 Tokenization

Tokenization is the process of splitting email body into sentences and words. Following figure shows how the system performs sentence level tokenization and word level tokenization. The NLTK library provides tokenizers for tokenization that are called sent_tokenization and word_tokenization tokenizer and they were imported to the python code in order to use them (Figure 6.10).

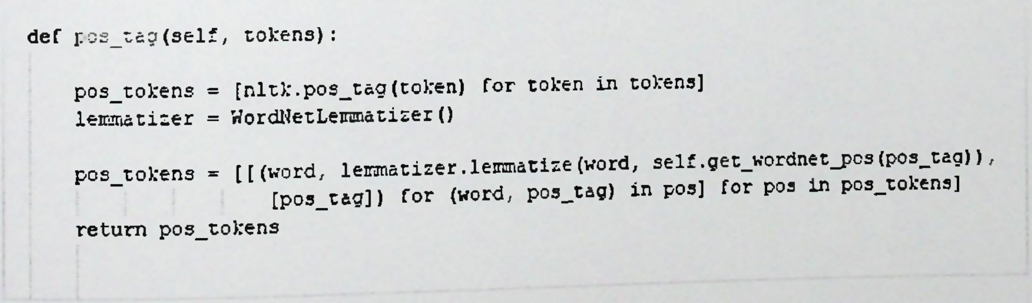


```
from nltk.tokenize import sent_tokenize, word_tokenize

def tokenize_sentences(self):
    tokenized_sentences = sent_tokenize(self.my_list)
    return tokenized_sentences

def tokenize_words(self, text):
    tokenized_words = word_tokenize(text)
    return tokenized_words
```

Figure 6.10: Source code for tokenizing text into sentences and words

6.7 Lemmatization

Lemmatization of words was performed by using WordNetLemmatizer and considering their part of speech tag (Figure 6.11). The accuracy can be increased in lemmatization if the POS tags are used during lemmatization process.



```
def pos_tag(self, tokens):

    pos_tokens = [nltk.pos_tag(token) for token in tokens]
    lemmatizer = WordNetLemmatizer()

    pos_tokens = [[(word, lemmatizer.lemmatize(word, self.get_wordnet_pos(pos_tag)),
                  [pos_tag]) for (word, pos_tag) in pos] for pos in pos_tokens]
    return pos_tokens
```

Figure 6.11: Lemmatization based on POS tags

6.8 Repetition handling

People sometimes write words abnormally.

Eg: Goooood, Hiiiii

The repetition of these additional consecutive characters should be removed in order to identify the word correctly. There are some letters that appear only once consecutively while there are some that only appear twice consecutively. The letters appear only once consecutively are ('h', 1), ('i', 1), ('j', 1), ('q', 1), ('w', 1), ('x', 1), ('y', 1) and all the other letters appear only twice in English words. The word "Goooood" has the letter 'o' repeating five times consecutively. Hence additional o's can be removed and only two o's can be kept in this word to fix it.

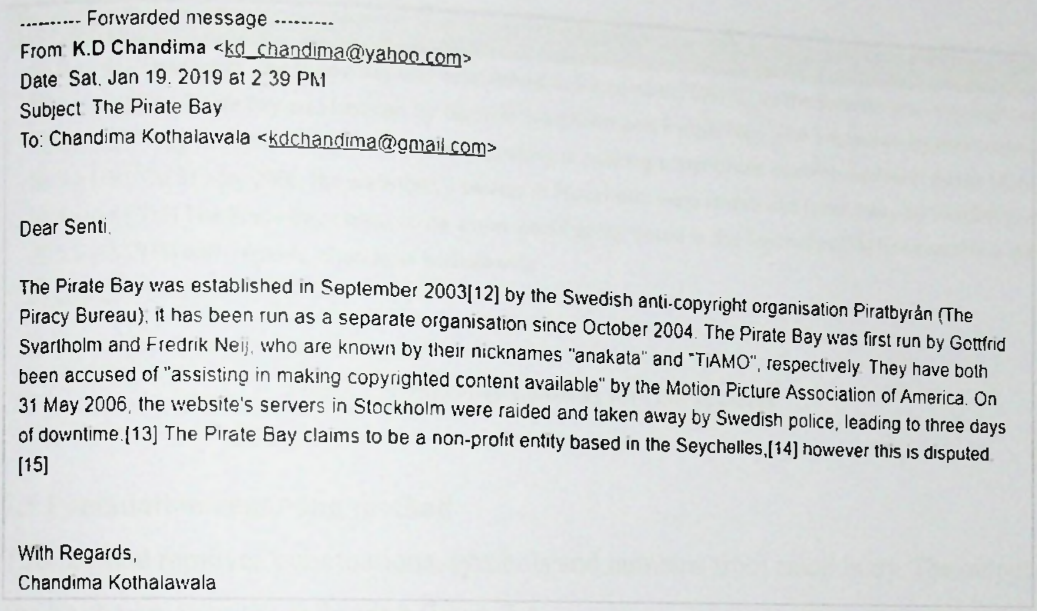
6.9 Idiom handling

Idioms were handled by maintaining a handmade idiom lexicon. For this project, 88 of most commonly and frequently used idioms were used and their meanings were stored in a separate text file. This idiom lexicon was used to replace the idioms found in the body text of emails by its relevant sentiment.

Evaluation and Results Chapter 7

7.1 Remove forwarded information method

This method removes the previously forwarded details of emails. An email with previously forwarded details can be seen in the figure 7.1.



---------- Forwarded message ---------
From K.D Chandima <kd_chandima@yahoo.com>
Date Sat, Jan 19, 2019 at 2 39 PM
Subject The Pirate Bay
To: Chandima Kothalawala <kdchandima@gmail.com>

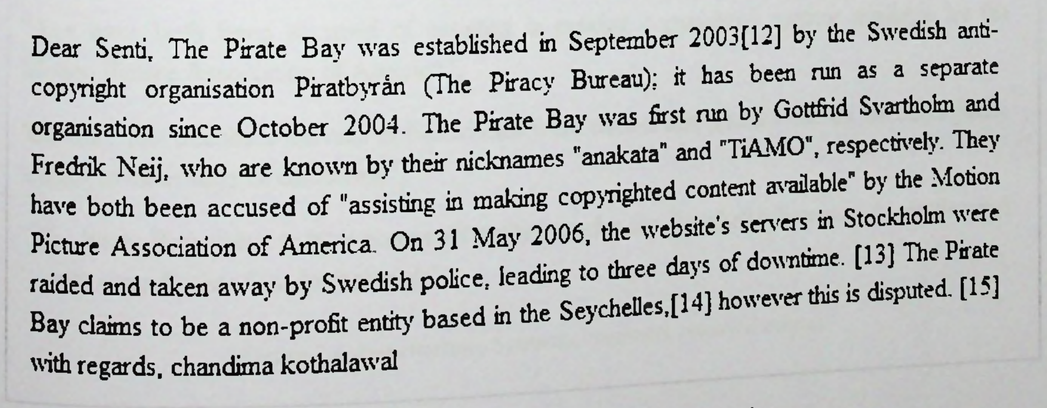
Dear Senti,

The Pirate Bay was established in September 2003[12] by the Swedish anti-copyright organisation Piratbyrån (The Piracy Bureau); it has been run as a separate organisation since October 2004. The Pirate Bay was first run by Gottfrid Svartholm and Fredrik Neij, who are known by their nicknames "anakata" and "TiAMO", respectively. They have both been accused of "assisting in making copyrighted content available" by the Motion Picture Association of America. On 31 May 2006, the website's servers in Stockholm were raided and taken away by Swedish police, leading to three days of downtime.[13] The Pirate Bay claims to be a non-profit entity based in the Seychelles,[14] however this is disputed.[15]

With Regards,
Chandima Kothalawala

Figure 7 1: E-mail body with previously forwarded information

After sending the email body text through this function, the output results can be shown in the figure 7.3 below. Notice that white spaces have been removed from the output by white space removing function.

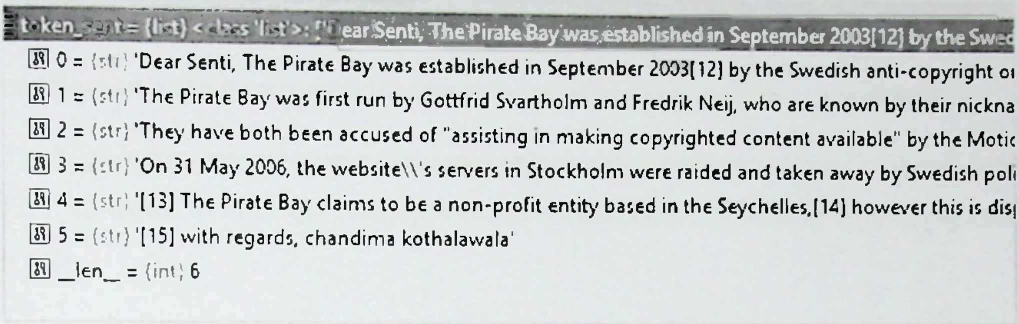


Dear Senti, The Pirate Bay was established in September 2003[12] by the Swedish anti-copyright organisation Piratbyrån (The Piracy Bureau); it has been run as a separate organisation since October 2004. The Pirate Bay was first run by Gottfrid Svartholm and Fredrik Neij, who are known by their nicknames "anakata" and "TiAMO", respectively. They have both been accused of "assisting in making copyrighted content available" by the Motion Picture Association of America. On 31 May 2006, the website's servers in Stockholm were raided and taken away by Swedish police, leading to three days of downtime. [13] The Pirate Bay claims to be a non-profit entity based in the Seychelles,[14] however this is disputed. [15] with regards, chandima kothalawal

Figure 7 2: After removing forwarded information

7.2 Tokenize sentences method

This method tokenizes the sentences of the email text. Notice that the phrase ‘Dear Senti’ has been attached to the first sentence while tokenizing (Figure 7.3). This means that the Tokenize sentence method is unable to differentiate the starting greeting and the first sentence. However, this discrepancy has an insignificant impact on the accuracy.

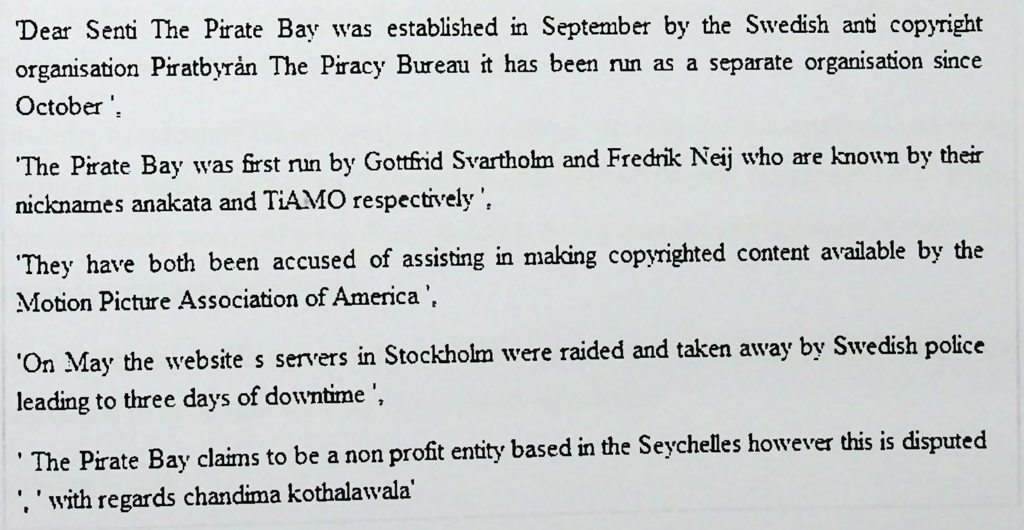


```
token_sent = {list} <class 'list'>: ['Dear Senti, The Pirate Bay was established in September 2003[12] by the Swed
 0 = {str} 'Dear Senti, The Pirate Bay was established in September 2003[12] by the Swedish anti-copyright o
 1 = {str} 'The Pirate Bay was first run by Gottfrid Svartholm and Fredrik Neij, who are known by their nickna
 2 = {str} 'They have both been accused of "assisting in making copyrighted content available" by the Motic
 3 = {str} 'On 31 May 2006, the website\\'s servers in Stockholm were raided and taken away by Swedish poli
 4 = {str} '[13] The Pirate Bay claims to be a non-profit entity based in the Seychelles,[14] however this is dis
 5 = {str} '[15] with regards, chandima kothalawala'
 __len__ = {int} 6
```

Figure 7 3: Tokenized sentences with line numbers

7.3 Punctuation removing method

This method removes punctuations, symbols and numbers from email body. The output can be shown as in the following figure 7.4.



```
'Dear Senti The Pirate Bay was established in September by the Swedish anti copyright
organisation Piratbyrån The Piracy Bureau it has been run as a separate organisation since
October ',

'The Pirate Bay was first run by Gottfrid Svartholm and Fredrik Neij who are known by their
nicknames anakata and TiAMO respectively ',

'They have both been accused of assisting in making copyrighted content available by the
Motion Picture Association of America ',

'On May the website s servers in Stockholm were raided and taken away by Swedish police
leading to three days of downtime ',

' The Pirate Bay claims to be a non profit entity based in the Seychelles however this is disputed
', ' with regards chandima kothalawala'
```

Figure 7 4: Punctuation, Symbols, Numbers removed output

7.4 Repetition removal method

The words like ‘Goooood’, ‘Cooool’ are fixed by the system using this method. The repetition of these additional consecutive characters should be removed in order to identify a word correctly. In words in English language there are some letters that appear only once consecutively while there are some letters that only appear twice consecutively. The letters appear only once consecutively are ('h', 1), ('i', 1), ('j', 1), ('q', 1), ('w', 1), ('x', 1), ('y', 1) and all the other letters may appear only twice in English words. The word “Goooood” has the letter ‘o’ repeating five times consecutively. The additional o’s can be removed leaving only two o’s in the word so as to correct it.

The logic behind consecutive repetition is revealed by examining an English dictionary [32]. However, it does not comply with compound words formed by combining two words.

For example,

Youthhood, washhouse, withhold, glowworm

Generally these words are written as **Youth hood**, **wash house** and **glow-worm**. They do not have a sentiment attached to them. Hence the logic accuracy of this method is high.

The system gives a corrupted output for the compound word **‘withhold’** when attempting to correct it using the above logic.

In-order to calculate the accuracy of this method, the logic behind repetition removing method has been checked on Electronic Pocket Oxford English Dictionary [33]. When this dictionary was used several words which do not comply with the logic of repetition removing method were found.

Among those words the words **“alibiing, genii, radii, taxiing, shanghaiing”** are converted to their base form by Lemmatization method.

alibiing = alibi

genii = genius

taxiing = taxi

radii = radius

shanghaiing = shanghai

Lemmatization process removes the ambiguity of these words and in addition there are no sentiments associated with these words. Hence the accuracy of the final output of this project would not make a difference.

Some other words, which are acronyms were also found in this dictionary and they are,

WWF - World Wide Fund for Nature

WWI - World War I

WWII - World War II

WWW - World Wide Web

ASCII

Again, there are no sentiment associated with these words and the accuracy of the final output of this project would not make a difference.

The following type of words were also found in the dictionary

Shiite, Shiism, skiing, burgundies, beachhead, powwow, hajj found from the dictionary and no sentiment is associated with those words.

Out of the total of nearly 140,000 words in the Pocket Oxford English Dictionary,

There are only 23 words which do not comply with the repetition removing method.

The accuracy of Repetition Removing Method can be denoted as

1-23/140,000*100 = 99.98%

7.5 Sentiment extraction from Lexicon based approach

Used combined lexicon

1. NRC Word-Emotion Association Lexicon (NRC Emotion Lexicon) - (14183 words with sentiments)

2. WordNet Affect Lexicon

Combined lexicon has more than 4000 words and 6 sentiments associated with them (anger, joy, sadness, obscene, negative and positive).

Sentiment extraction was started by creating bags of words from the combined lexicon. They are called happy-bag, sad-bag, defamatory-bag, obscene-bag, negative-bag and positive-bag.

Word count of each word bag

Obscene-bag = 445

Happy-bag = 1023

Sad-bag = 1195

Defamatory-bag = 1476

Tokenized words are then checked from these bags to find out the sentiment. A sentiment count is taken for words in each sentence as happy count, sad count, defamatory count and obscene count. The sentiment with the highest probability becomes the sentiment of the sentence. Sentiments of all sentences are then considered to determine the overall sentiment for a given email.

7.6 Sentiment extraction from Machine Learning Approach

Training dataset has 100 Defamatory emails, 200 Happy emails, 150 Sad emails, 7 Obscene emails and 40 Complex emails which are difficult to assess the sentiment directly.

Testing dataset has 14 Defamatory emails, 31 Happy emails, 27 Sad emails, 3 Obscene emails and 23 Complex emails which are difficult to assess the sentiment directly.

In the sentiment extraction process sentence level classification has been used, and this involves labelling of each sentence of each email to create training and testing dataset. Training dataset has nearly 3000 sentiment labelled sentences and testing dataset has nearly 750 sentiment labelled sentences.

Five sentiments are input to the neural network as five features. Machine learning approach has an additional sentiment category compared to Lexicon based approach, and this category only contains the word 'not'. If there are two not words in a sentence then value of not category becomes 2.

not = 1 not[not, not] becomes 2

If one category of features eg. nouns has the value 0 or no words in it, then the value assigned is 0.

The value 0 is considered as a neutral sentiment. The extracted features are fed into a neural network and the neural network outputs five probability values corresponding to five sentiments.

7.7 Classifier Model

The classifier model was created by training the neural network using 3000 labeled sentences. Then the predictions were made based on this classifier model for five classes.

7.8 Accuracy

Testing was done using 98 emails and the confusion matrix (Figure 7 5: Confusion Matrix for Machine Learning Approach) has been created to indicate the accuracy.

Lexicon based approach accuracy = 70%

Machine learning based approach accuracy = 37.75%

	Happy	Sad	Defamatory	obscene	Neutral	
Happy	14	10	8	3	0	35
Sad	12	8	4	6	0	30
Defamatory	4	4	6	0	0	14
obscene	1	1	1	5	0	8
Neutral	2	4	1	0	4	11
	33	27	20	14	4	N =98

Figure 7 5: Confusion Matrix for Machine Learning Approach

Chapter 8

Discussion

8.1 Discussion

Text preprocessing, understanding the sentiment and classification by detecting sentiment belongs to the e-mail classification process. Different methodologies, techniques and tools are available to analyze texts and extract the meaning of texts. These methods, techniques and tools provide different features and different levels of accuracy. Choosing a set of methodologies, techniques and tools to a particular task should be done by performing a careful study about them.

Lack of data and lack of classification levels provides low accuracy in sentiment analysis. Analyzing a large amount of data leads to decreased performance of such systems. Most of the sentiment analysis tasks are done real-time and hence, analyzing a large amount of data will make a considerable impact on the performance of sentiment analysis.

Sentiments are very likely to change over time according to a person's mood, world events, and so forth. Sarcasm and ironical language are difficult for text analyzing algorithms to determine the sentiment accurately.

8.2 Conclusion

An e-mail may contain a happy news or a sad news or defamatory content which might make the reader happy, sad or distressed. Sometimes the e-mails may contain obscene quotes and pictures. If there is a way for a user to get an idea of the type of content in the email prior to opening them, then he or she can choose which e-mails should be opened first.

A proper e-mail categorization is required and this attempt is to categorize e-mails according to the sentiment and the ingredient of the e-mail content, by indicating emoticons on the subject of e-mails.

A lexicon based and a Machine learning based approach are used in this project to extract the sentiment from each sentence of the email body. The accuracy of machine learning method seems to provide more accurate results compared to lexicon based method when the training dataset is large. Different features provide different accuracy levels in neural networks. Extracting proper features set is important to have better results in machine learning.

Negation handling has been introduced as a feature to increase the accuracy of machine learning approach. Number of not words have been selected as a feature for the neural network.

It seems sentence level sentiment extraction gives some difficulties the neural network to properly identify the sentiments for a whole email. Document level sentiment extraction can be suggested to increase the accuracies.

8.3 Future works

The idiom lexicon used in this project has 88 chosen idioms and the relevant sentiment. Increasing the number of idioms and introducing phrases in English language to this list will lead to a greater accuracy in sentiment analysis. There are at least twenty-five thousand idiomatic expressions in the English language [34].

The combined lexicon used in this project has 4139 of words from the four sentiments; angry, happy, sad, and obscene. Increasing the number of words in the lexicon can increase the accuracy in sentiment extraction.

The adult web site domain list used in this project has 109 domain names and increasing the number of domain names can help the systems to provide a more accurate result.

The performance of a text classification model is heavily dependent upon the type of words used in the corpus and type of features created for classification. In some cases, features as the combination of words provide a better significance than considering single words as features. Combination of N number of words together are called N-grams. It is known that bigrams are the most informative N-Gram combinations. Adding bigrams to a feature set will improve the accuracy of the text classification model.

Labeling of training data (for each sentence) was done manually. To have more accurate result, the data set has to be labeled by several people and an average has to be taken.

It seems sentence level sentiment extraction creates some difficulties in identification of sentiments for a whole email using the neural network. It is suggested that a document level sentiment extraction can be used to increase accuracy.

References

[1] M. A. Javed and C. Technology, "Numerical Optimisation of the Learning Process."

[2] W. Z. Khan, M. K. Khan, F. T. Bin Muhaya, M. Y. Aalsalem, and H. C. Chao, "A Comprehensive Study of Email Spam Botnet Detection," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2271–2295, 2015.

[3] P. Rajendran, M. Janaki, S. M. Hemalatha, and B. Durkananthini, "Adaptive privacy policy prediction for email spam filtering," *IEEE WCTFTR 2016 - Proc. 2016 World Conf. Futur. Trends Res. Innov. Soc. Welf.*, 2016.

[4] K. Patel and S. K. Dubey, "To recognize and analyze spam domains from spam emails by data mining," pp. 4030–4035, 2016.

[5] I. Idris, A. Selamat, and S. Omatu, "Hybrid email spam detection model with negative selection algorithm and differential evolution," *Eng. Appl. Artif. Intell.*, vol. 28, pp. 97–110, 2014.

[6] I. Idris and A. Selamat, "Improved email spam detection model with negative selection algorithm and particle swarm optimization," *Appl. Soft Comput. J.*, vol. 22, pp. 11–27, 2014.

[7] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Syst.*, vol. 64, pp. 22–31, 2014.

[8] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," *Expert Syst. Appl.*, vol. 41, no. 4 PART 2, pp. 1937–1946, 2014.

[9] J. Clark, I. Koprinska, and J. Poon, "Linger - A Smart Personal Assistant for E-Mail Classification," *Proc. 13th Int. Conf. Artif. Neural Networks*, pp. 274–277, 2003.

[10] M. K. Chae, A. Alsadoon, P. W. C. Prasad, and S. Sreedharan, "Spam filtering email classification (SFECM) using gain and graph mining algorithm," *2017 2nd Int. Conf. Anti-Cyber Crimes, ICACC 2017*, pp. 217–222, 2017.

[11] S. Bin Abd Razak and A. F. Bin Mohamad, "Identification of spam email based on information from email header," *Int. Conf. Intell. Syst. Des. Appl. ISDA*, pp. 347–353, 2014.

[12] "AN ONTOLOGY-BASED APPROACH TO TEXT SUMMARIZATION," 2017. [Online]. Available: http://myprojectbazaar.com/product/final-projects/ontology-based-approach-text-summarization-www-myprojectbazaar-com/. [Accessed: 11-Sep-2017].

[13] S. Aridhi and E. Mephu Nguifo, "Big Graph Mining: Frameworks and Techniques," *Big Data Res.*, vol. 6, pp. 1–10, 2016.

[14] S. Goswamil and J. Poray, "Human computer interaction for sentiment analysis and opinion mining: A review," *2016 Int. Conf. Comput. Electr. Commun. Eng. ICCECE 2016*, 2017.

[15] Y. M. Aye, "6Hqwlphqw $ Qdo \ Vlv Iru 5Hylhzv Ri 5Hvwdxudqwv Lq 0 \ Dqpdu 7H [W," pp. 321–326, 2017.

[16] H. Kaur, V. Mangat, and Nidhi, "A Survey of Sentiment Analysis techniques," *Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud) (I-SMAC 2017)*, pp. 921–925, 2017.

[17] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Comput. Intell.*, vol. 29, no. 3, pp. 436–465, 2013.

[18] A. Valdivia, M. V. Luzón, and F. Herrera, "Sentiment Analysis in TripAdvisor," *IEEE Intell. Syst.*, vol. 32, no. 4, pp. 72–77, 2017.

[19] M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," *IEEE Access*, vol. 3536, no. c, pp. 1–21, 2017.

[20] C. Dedhia and J. Ramteke, "Ensemble model for Twitter Sentiment Analysis," pp. 1–5, 2017.

[21] D. Stojanovski, "Twitter Sentiment Analysis using Deep CNN," vol. 9121, no. JUNE, 2015.

[22] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06*, 2006, p. 43.

[23] I. I. Conference, "SELECT-ADDITIVE LEARNING : IMPROVING GENERALIZATION IN MULTIMODAL SENTIMENT ANALYSIS Haohan Wang , Aaksha Meghawat , Louis-Philippe Morency and Eric P . Xing Language Technologies Institute School of Computer Science Carnegie Mellon University," no. July, pp. 949–954, 2017.

[24] M. Sammons *et al.*, : "Feature Extraction for NLP, Simplified," pp. 4085–4092, 2015.

[25] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha, and S. Yenduri, "Application of Deep Learning to Sentiment Analysis for recommender system on cloud," *2017 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 93–97, 2017.

[26] C. William W. Cohen, MLD, "Enron Email Dataset," 2015. [Online]. Available: https://www.cs.cmu.edu/~./enron/. [Accessed: 03-Feb-2018].

[27] J. Bartholomew, "30 Angry Letters," 2010. [Online]. Available: https://www.austinchronicle.com/year-thirty/lists/thirty-angry-letters/. [Accessed: 03-Mar-2018].

[28] "The Millenium Project." [Online]. Available: http://www.ratbags.com/rsoles/files/mailbox.htm. [Accessed: 03-Feb-2018].

[29] S. M. Mohammad, "Sentiment and Emotion Lexicons." [Online]. Available: http://saifmohammad.com/WebPages/lexicons.html. [Accessed: 24-Mar-2018].

[30] H. R. Unit, "WordNet Domains," 2009. [Online]. Available: http://wndomains.fbk.eu/wnaffect.html. [Accessed: 10-Jun-2018].

[31] R. Grishman, "Natural language processing," *J. Assoc. Inf. Sci. Technol.*, vol. 35, no. 5, pp. 291–296, 1984.

[32] T. . R. Rog, "ROGET'S THESAURUS OF ENGLISH WORDS AND PHRASES." NEW YORK THOMAS Y. CROWELL COMPANY, p. 532, 1911.

[33] *Electronic Pocket Oxford English Dictionary*, vol. 7. Oxford University Press, Great Clarendon Street, Oxford OX2 6DP, 2002.

[34] "Idiom," 2018. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Idiom&oldid=874927067.

[Accessed: 15-Dec-2018].