A Major project Report on

Emotion Recognition From Text

Submitted in partial fulfilment for the
degree of Bachelor of Technology in
Information Technology

Submitted by
Nimisha Jadav (1613013)
Komal Shirsat (1613038)
V.Preethi (1613049)

Under the guidance of
Prof. Mohan Bonde

Usha Mittal Institute of Technology
S.N.D.T. Women's University
Juhu Tara Road, Santacruz(W), Mumbai-400049
2019-2020

CERTIFICATE

This is to certify that Nimisha Jadav(1613013), Komal Shirsat(1613038) and V.Preethi (1613049) has completed the Major project report on the topic "Emotion Recognition From Text" in partial fulfilment for the Bachelor's Degree in Information Technology under the guidance of Prof. Mohan Bonde during the year 2019-2020 as prescribed by SNDT Women's University.

Guide	Head Of Department
Prof. Mohan Bonde	Dr. Sanjay Shitole

Principal
Dr. Sanjay Pawar

Examiner 1	Examiner 2

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Nimisha Jadav (1613013)
Komal Shirsat (1613038)
V.Preethi (1613049)

Date

Abstract

Human Computer interaction is a very powerful and most current area of research because the human world is getting more digitised. This needs the digital systems to imitate the human behaviour correctly. Emotion is one aspect of human behaviour which plays an important role in human computer interaction, the computer interfaces need to recognize the emotion of the users in order to exhibit a truly intelligent behaviour.There is still a question on how to detect emotion from a text input. To solve this problem, this project generates an Emotion Recognition Model to extract emotion from text at the sentence level. Our method detects emotion from a text-input by using different deep learning algorithms like CNN, LSTM, SVM and HAN. The experiments show that the method could generate a good result for emotion detection from text input. To recognize emotion from text we have considered six emotions class such as joy, sadness, anger, love, fear, surprise).
Keywords: Emotion recognition ,NLP, CNN, LSTM, SVM, HAN

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emotion is one type of affect, other type of being mood, temperament and sensation. Emotions have been widely studied in psychology and in behaviour sciences, as they are an important element of human nature.

Our emotions influence every aspect of our lives, from how we connect with each other, to how we make decisions, to our health and well-being. Your emotional state can affect a very simple decision, like what you're having for breakfast, to very big decisions, like what house you're going to buy or who you're going to marry. People who have high EQ, who are able to translate their emotion reading and sensing skills into their day-to-day behavior, tend to be more likable as human beings. They tend to be more successful in their professional lives and they actually tend to be healthier. They live longer and happier lives in general.

Nowadays it is important in the field of human computer interactions. Emotions are also an important aspect in the interaction and communication between people. The exchange of emotions through text messages and posts of personal blogs poses the informal style of writing challenge for researches. Extraction of emotions from text can be applied for deciding the human computer interaction which governs communication and many more.

Emotions may be expressed by a person's speech, facial expressions and text based emotions respectively. Emotions may be expressed by one word or a bunch of words. Sentence level emotion detection method plays a crucial role to trace emotions or to search out the cues for generating such emotions. Sentences are the essential information units of any document. For that reason, the document level emotion detection method depends on the emotion expressed by the individual sentences of that document that successively relies on the emotions expressed by the individual words.

In computational linguistics, the recognition of emotions from texts is becoming crucial from an application point of view. The examples are affective computing, the tasks of opinion mining and market analysis, or natural language interfaces such as e-learning environments or educational and edutainment games. This needs computational approaches to successfully analyse this online content, recognize, and draw useful conclusions and detection of emotions.

1.1 Review of Literature

1. **Semantic-Emotion Neural Network for Emotion Recognition From Text**

   This paper explored an emotion recognition method from text based on the combined network which consists of CNN based emotion encoder and BiLSTM based semantic encoder called SENN, a novel model is proposed and applied on ten real-world datasets.For the SENN model, BiLSTM is designed to capture contextual information and CNN is designed to extract emotional information effectively.

   Experimental results show that the SENN model outperforms most of the baseline methods and state-of-the-art approaches. Compared with traditional machine learning models, authors proved that deep learning based models outperformed the machine learning models as reported in previous studies. Logistic regression and support vector machine shows the comparative result using bag-of-word and tf-idf vectors. Compared with the state-of-the-art models in emotion classification, SENN gives the best performance on nine out of ten datasets except Tales-Emotion dataset. It performs F1-scores of 84.8%, 51.1%, 61.3%, 74.6%, 91.0%, 56.3%,59.3%, 98.8% and 70.8% on real-world datasets. And CNN gives the best result on Tales-Emotion dataset using FastText word embedding.[2]

2. **Classification Model To Determine The Polarity Of Movie Review Using Logistic Regression**

   In this paper, feature extraction has been done using bag of words specifically bi-grams, which has powerful impact on determining the polarity of the movie review. The model is then trained using logistic regression machine learning classification algorithm , which is showing the accuracy of 88% .When a new set of reviews are feed into the model the model will predict the polarity of the movie reviews. This can enhance the feature extraction process so that model can have more precise features and also by applying different machine learning classification algorithm, accuracy of the model can be improved.[5]

3. **Emotion Analysis: A Survey**

   This survey paper is based on prior works done in the field of emotion analysis through text which is an emerging field with many applications in real world. There has been a lot work done in the field from past and the researches are still on, particularly, using the Tweet data. However, text emotion analysis also introduces some challenges in work in the sense that emotions and the ways to express these emotions are all subjective. The emotion analysis uses the natural language processing, text analysis and various computational techniques to determine the emotions hidden in a particular text. This analysis can be done at various levels: document level , sentence level, word level, and aspect level [4]

4. **EmoTxt: A toolkit for emotion recognition from text**

   The system is completely developed in Java and distributed under the MIT open-source license . Author has release the classification models trained on gold standard datasets, which can be used for emotion detection from text. EmoTxt identifies emotions in an input corpus provided as a comma separated value (CSV) file, with one text per line, preceded by a unique identifier. The output is a CSV file containing the text id and the predicted label for each item of the input collection.Other than classification, EmoTxt supports training of emotion classifiers from manually annotated training data. Its training approach leverages a suite of features that are independent of the theoretical model adopted for labeling the data.[3]

5. **A Corpus for Sentiment Analysis and Emotion Recognition for a Learning Environment**
   In this paper, author has describe the development of a system to generate a corpus of textual opinions in Spanish, labeled with learning-centered emotions. The corpus generated with the ERAS system contains 851 textual opinions. The system remains available for more participants to express their opinions on educational resources. This model will help Intelligent Tutoring Systems to detect emotions through text and make the teaching process more efficient for students, adjusting the content to the particular needs of each of them. [6]

6. **Emotion Recognition from Text Based on Automatically Generated Rules**
   The problem of emotion recognition or emotion detection from text to the problem of finding relations between the input sentence and the emotional content within it. Intuitively, finding these relations relies on discovering specific terms (emotional keywords, verbs, nouns, etc.) in the input sentence and other deeper inferences that are related to the meaning of the sentence. Once these terms and their relation to the meaning of the sentence are found, they can be generalized and considered as emotion recognition rules (ERRs). For example, consider the sentence "I received many gifts on Christmas Eve"; Assuming that this sentence reflects a happy emotion, by analyzing the sentence we can reach to the conclusion that the verb"received" and the noun "gifts" are the most important parts of the sentence, and consequently we can come up with a rule that says "receiving gifts" reflects the emotion happy. [7]

7. **Emotion Detection From Text Documents**
   This model is based on keyword spotting technique, apart from that it also uses the concept of ontology. Use of ontology makes this model more efficient than other methods in recognizing emotions from text input. This has been created to overcome following limitations:

   - Ambiguity in Keyword Definitions: The meanings of keywords could be multiple and vague, as most of the words could change their meanings according to different usages and contexts.

- Incapability of Recognizing Sentences without Keywords: "I passed my qualify exam today" and "Hooray! I passed my qualify exam today" should imply the same emotion (joy), but the former sentence without "hooray" could remain undetected if "hooray" is the only keyword to detect this emotion.
- Lack of Linguistic Information: Syntax structures and semantics also have influences on expressed emotions. For example, "I laughed at him" and "He laughed at me" would suggest different emotions from the first person's perspective. As a result, ignoring linguistic information also poses a problem to keyword-based methods.
- Difficulties in Determining Emotion Indicators Learning-based methods can automatically determine the probabilities between features and emotions but the methods still need keywords in the form of features. The most intuitive features may be emoticons which can be seen as author's emotion annotations in the texts. [8]

8. **Text Based Emotion Recognition: A Survey**
This paper is mainly focused on an overview of emotion detection from text and describes the emotion detection methods. These methods are divided into the following four main categories: keyword-based, Lexical Affinity method, learning based, and hybrid based approach. Limitations of these emotion recognition methods are presented in this paper and also, addresses the text normalization using different handling techniques for both plain text and short messaging language.This approach is easy to implement and intuitive since it involves identifying words to search for in text. These words are classified into categories such as disgust, sadness, happy, anger, fear, surprise etc.Existing system make use of plain text only. This paper describes the different text based emotion recognition methods and their limitations. The problems are faced by the emotion recognition system while processing raw text which contain both plain text and short messaging language. This paper addresses the existing different approaches for resolving processing of

raw textual data which contain combination of both plain text and short messaging language.[1]

1.2 Problem Statement

In today's internet world, human expresses their emotions,sentiments, feelings via text or comments, emojis, likes and dislikes.Understanding the true meanings behind these electronic text is very crucial.
The available approaches are work in the direction of recognizing the polarity of sentiment. The sentiment maybe positive or negative. Among the less explored sentiment areas is the recognition of types of emotions from text documents. Recognizing emotions conveyed by a text can provide an insight into the author's intent and may lead to better understanding of the text's content.

1.3 What is Emotion Recognition?

Emotion Recognition system can accurately detect the emotion from any textual data. People voice their opinion, feedback and reviews on social media, blogs and forums. Marketers and customer support can leverage the power of Emotion Detection to read and analyze emotions attached with the textual data.

There are 6 emotion categories that are widely used to describe humans' basic emotions, based on facial expression anger, disgust, fear, happiness, sadness and surprise.

In any recognition task, the 3 most common approaches are rule-based, statistic-based and hybrid, and their use depends on factors such as availability of data, domain expertise, and domain specificity. In the case of sentiment analysis, this task can be tackled using lexicon-based methods, machine learning, or a concept-level approach. Here, we are exploring how we can achieve this task via a machine learning approach, specifically using the deep learning technique.

One of the biggest challenges in determining emotion is the context-dependence of emotions within text. A phrase can have element of anger without using the word "anger" or any of its synonyms. For example, the phrase "Shut up!". Another challenge is the difficulty that other components of NLP are facing, such as word-sense disambiguation and co-reference resolution. It is difficult to anticipate the success rate of machine learning approach without first trying.

Initially, texts from a variety of articles in this wide range of sources will be extracted. However, there is no guarantee that the text will provide a balanced mix of sentences with all the necessary categories of emotions. In other words, since not all sentences will contain emotional cues that we are interested in, it is highly possible that the proportion of “neutral” sentences outweigh those with emotional elements.

1.4 Scope of Project:

Sufficient amount of work has been done related to speech and facial emotion detection but text based emotion recognition system still requires attraction of researchers. The short messaging language have the ability to interrupt and falsify Natural language processing tasks done on text data.

To illustrate that ability, consider an example, "At de moment he cnt just put me in da better zone though. happy bday mic, ur a legend".

At this moment when going through this sentence, it will recognize some terms which doesn't belong to decent plain text. But while going through these sentences, then and there human brain will resolve the short messaging language word to a meaningful word or phrase. When human see "cnt" and its neighboring words "he" and "just", human know that it is "can't". That's because human brain is trained with previous experiences. But when it comes to Natural Language Processing tools, they are trained and adopted to work properly with plain text. Mapping short messaging language words to plain text words can be very sensitive at some cases.

A wrong mapping can result in alternations of the meaning or it may destroy semantics under the applied context. When considering the sub phrase "ur a legend" in above example, "ur" can be considered as "your" or "you are". Humans can understand that its "you're a legend" and not "your a legend". But a direct mapping from a language tool would not. Hence it depends on the context which the word is used.

1.5 Application areas of Textual Emotion Detection

1. **Sentiment Analysis** :

   Sentiment Analysis mainly focuses on information retrieval and knowledge discovery from text. Sentiment Analysis is also called as opinion mining . Opinion are often collected web forums, blogs, discussion groups, comment boxes and online e-learning systems. Opinion Mining is one amongst a vital application of web data. It is used to collect user opinion and extract meaningful patterns from it.

   Sentiment analysis is used for understanding the consumer, improving perception of a customer with the ultimate goal to increase brand reputation and sales.

   Business analysts, product managers, customer support directors, human resources and workforce analysts, and other stakeholders use sentiment analysis to understand how customers and employees feel about particular subjects, and why they feel that way.

2. **Text–to–Speech Generation**:

   The goal of text to speech generation is to classify the emotional resemblance of sentences in the text, for detecting exact expressive representation of text-to- speech synthesis. Text documents are just collection of sentences which have emotional content. In verbal communication, readers should express the exact emotions from that text by modifying manner of speech, including pitch, intensity etc. So, in order to generate text - to – speech conversion, it is important to recognize the emotions from the text documents.

3. **Improves Human computer interaction** :

   The emotion recognition system should be applied in different kinds of the Human computer interaction systems, such as dialogue systems, automatic answering systems and human robots etc. A system that is based on the user's emotion,makes human computer interaction synchronized.

4. **Happiness Indexes**:

   Since the mid-2000s, Government and organizations around the world are paying more attention to the happiness index.

   The Obama administration used sentiment analysis to gauge public opinion to policy announcements and campaign messages ahead of 2012 presidential election

   The Happy Planet Index (HPI): This metric is defined as the overall index scores that rank countries based on their efficiency, as well as how many long and happy lives each country produces per unit of environmental output. This is unusual because the majority of indexes are based upon economic measures.

   Societal Wellbeing metrics: The UK government measures people's well-being; their statistics can be found here. Other countries and cities such as Seattle, Dubai, and South Korea, have similar measures.

Chapter 2

Hardware and software requirements

2.1 Software Requirements:

2.1.1 Python

Python is a very simple and easy to learn general purpose programming language. Python is the most popular language, because of its advantages like, readability, portability, open source structure etc.Python is used to develop different applications.Python is also used for Machine Learning. NumPy, Pandas and Matplotlib libraries are used for Data Manipulation and Visualization.Python is also used for Scripting. Scripting is used to automate complicated tasks easily.Python is an open source programming language that was made to be easy-to-read and powerful.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly. Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it.Writing programs in Python takes less time than in some other languages.

2.1.2 Spyder: The Scientific Python Development Environment

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It offers a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing developers to build upon its functionality and embed its components, such as the interactive console, in their own PyQt software.

2.1.3 The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It provides an environment, where you can document your code, run it, look at the outcome, visualize data and see the results without leaving the environment. This makes it a handy tool for performing end to end data science workflows – data cleaning, statistical modeling, building and training machine learning models, visualizing data, and many other uses.

Jupyter Notebooks really shine when you are still in the prototyping phase. This is because your code is written in independent cells, which are executed individually. This allows the user to test a specific block of code in a project without having to execute the code from the start of the script.

2.1.4 Natural Language Processing and Natural Language Toolkit

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken. NLP is a component of artificial intelligence

(AI).Syntax and semantic analysis are two main techniques used with natural language processing. Syntax techniques used include parsing, word segmentation, sentence breaking, morphological segmentation and stemming. Semantics involves word sense disambiguation (which derives meaning of a word based on context), named entity recognition (which determines words that can be categorized into groups), and natural language generation .

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports. NLTK includes more than 50 corpora and lexical sources such as the Penn Treebank Corpus, Open Multilingual Wordnet, Problem Report Corpus, and Lin's Dependency Thesaurus.

2.1.5 Deep learning algorithms:

A machine learning or deep learning model is able to generalize and deal with novel cases. If a case resembles something the model has seen before, the model can use this prior "learning" to evaluate the case. The goal is to create a system where the model continuously improves at the task you've set it.

Deep learning for NLP and text analytics involves a set of statistical techniques for identifying parts of speech, entities, sentiment, and other aspects of text.

The techniques can be expressed as a model that is then applied to other text, also known as supervised machine learning. It also could be a set of algorithms that work across large sets of data to extract meaning, which is known as unsupervised machine learning.

The most popular NLP machine learning or deep algorithms which we have used are as follows:

1. **Long Short Term Memory**:

   Long Short-Term Memory (LSTM) networks are a type of recurrent neural

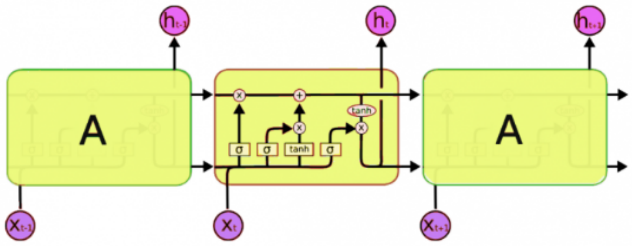


Figure 2.1: LSTM architecture

network (RNN) capable of learning order dependence in sequence prediction problems. Recurrent neural networks are different from traditional feed-forward neural networks..In RNN output from the last step is fed as input in the current step. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficient performance. LSTM can by default retain the information for long period of time. It is used for processing, predicting and classifying on the basis of time series data. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

The two technical problems overcome by LSTMs are vanishing gradients and exploding gradients, both related to how the network is trained.

An LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. These blocks can be thought of as a different version of the memory chips in a digital computer. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates that provide continuous analogues of write, read and reset operations for the cells and control information flow drawing on the logistic function.

Apple, Amazon, Google, Microsoft and other companies incorporated LSTM as a fundamental element into their products.

2. **Convolutional Neural Network**:

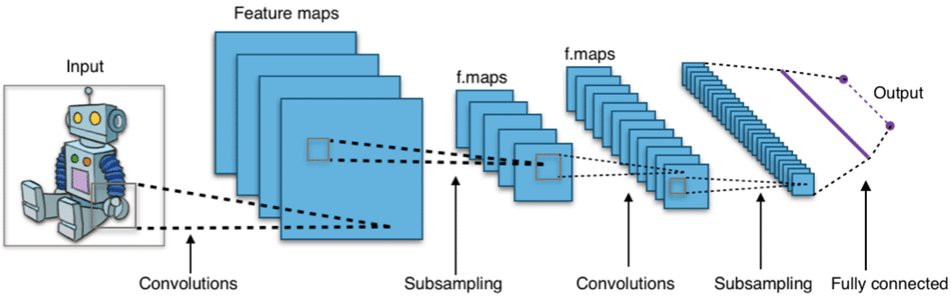


Figure 2.2: CNN model structure

CNN is a class of deep, feed-forward artificial neural networks (where connections between nodes do not form a cycle) and use a variation of multi-layer perceptrons designed to require minimal preprocessing. It can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

A convolutional neural network (CNN) contains one or more convolutional layers, pooling or fully connected, and uses a variation of multi-layer perceptrons discussed above. Convolutional layers use a convolution operation to the input passing the result to the next layer. This operation allows the network to be deeper with much fewer parameters.

Convolutional neural networks show outstanding results in image and speech applications..In Text Understanding from Scratch, CNNs can achieve outstanding performance without the knowledge of words, phrases, sentences and any other syntactic or semantic structures with regards to a human language . Semantic parsing , paraphrase detection, speech recognition are also the applications of CNNs.

3. **Support Vector Machine**:

   Support Vector Machine(SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.

   The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

   To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

   Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.

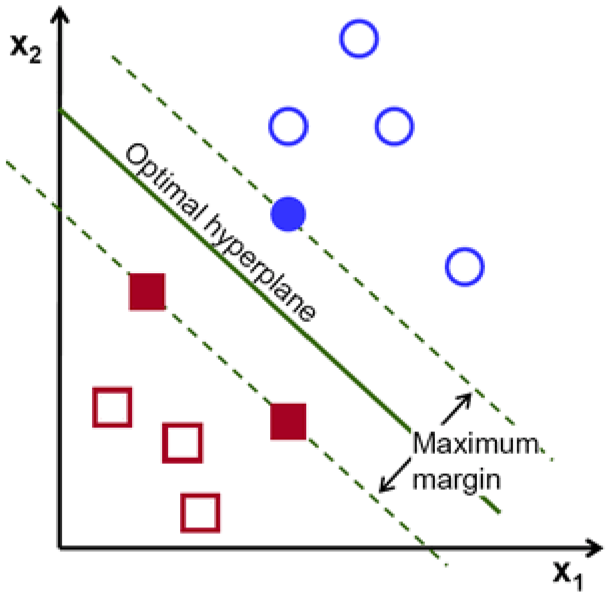


Figure 2.3: SVM

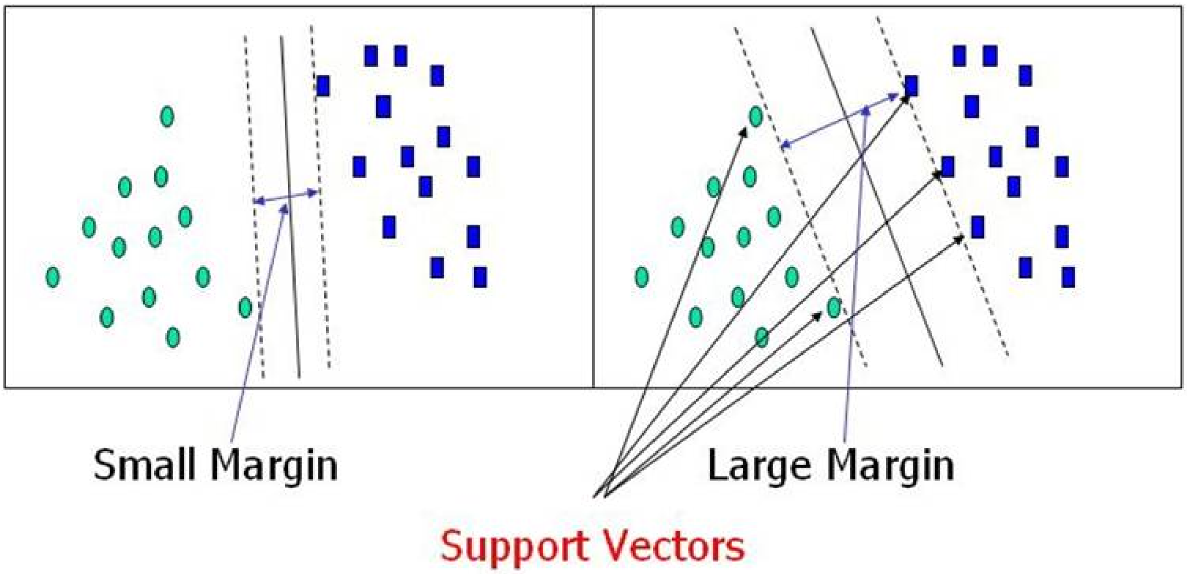


Figure 2.4: SVM

4. **Hierarchical Attention Network (HAN)**:

   The overall architecture of the Hierarchical Attention Network (HAN) is shown in Figure.

   It uses stacked recurrent neural networks on word level followed by attention model to extract such words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector. Then the same procedure applied to the derived sentence vectors which then generate a vector who conceives the meaning of the given document and that vector can be passed further for text classification.

   Attention model consists of two parts: Bidirectional RNN and Attention networks. While bidirectional RNN learns the meaning behind those sequence of words and returns vector corresponding to each word, Attention network gets weights corresponding to each word vector using its own shallow neural network. Then it aggregates the representation of those words to form a sentence vector i.e it calculates the weighted sum of every vector. This weighted sum embodies the whole sentence. The same procedure applies to sentence vectors so that the final vector embodies the gist of the whole document. Since it has two levels of attention model, therefore, it is called hierarchical attention networks.

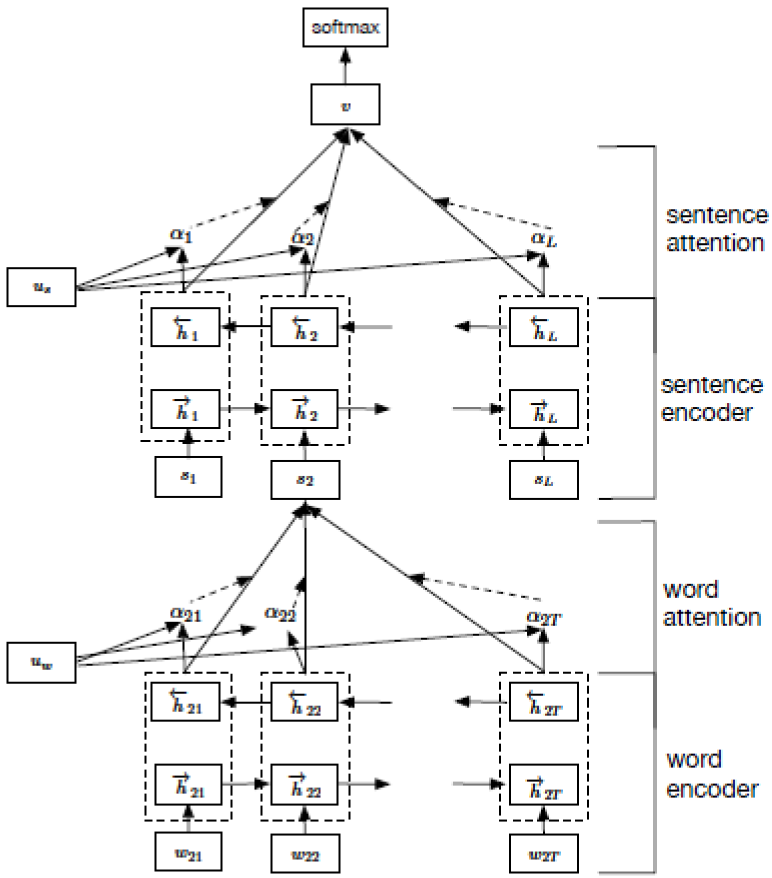


Figure 2.5: HAN

2.1.6 Tweepy

Twitter is a popular social network where users share messages called tweets. Twitter allows us to mine the data of any user using Twitter API or Tweepy. The data will be tweets extracted from the user.

The first thing to do is get the consumer key, consumer secret, access key and access secret from twitter developer available easily for each user. These keys will help the API for authentication. Tweepy is a tool to access Twitter data in a fairly easy way with Python. There are different types of data we can collect, with the obvious focus on the "tweet" object.

Once we have collected some data, the possibilities in terms of analytics applications are endless. One such application of extracting tweets is sentiment or emotion analysis. The emotion of the user can be obtained from the tweets by tokenizing each word and applying machine learning algorithms on that data. Such emotion or sentiment detection is used worldwide and will be broadly used in the future.

Chapter 3

Methodology and Implementation of the proposed system

Emotion classification can be divided into two different categories:
1.coarse-grained level (positive or negative) which can be accurately perceived from text.
2.fine grained level(the six Ekman emotions) it requires semantic and syntactic analysis of the sentence and can be done using three methods viz.,Keyword-based detection,Learning-based detection, and Hybrid detection.

1. Keyword-based detection : classifying emotions is done by searching for the emotional keywords in the input sentence .Such methods suffer from the ambiguity in the keyword definitions in the sense that a word can have different meanings according to usage and context, the incapability of recognizing emotions within sentences that do not contain emotional keywords.

2. Learning-based detection: In these methods, the emotion is detected by using classification approaches based on a training dataset.

3. Hybrid detection: In hybrid methods, emotions are detected by using a combination of emotional keywords and learning patterns collected from training datasets, in addition to information from different sciences, like human psychology, system starts by looking for emotional abbreviations and emoticons.

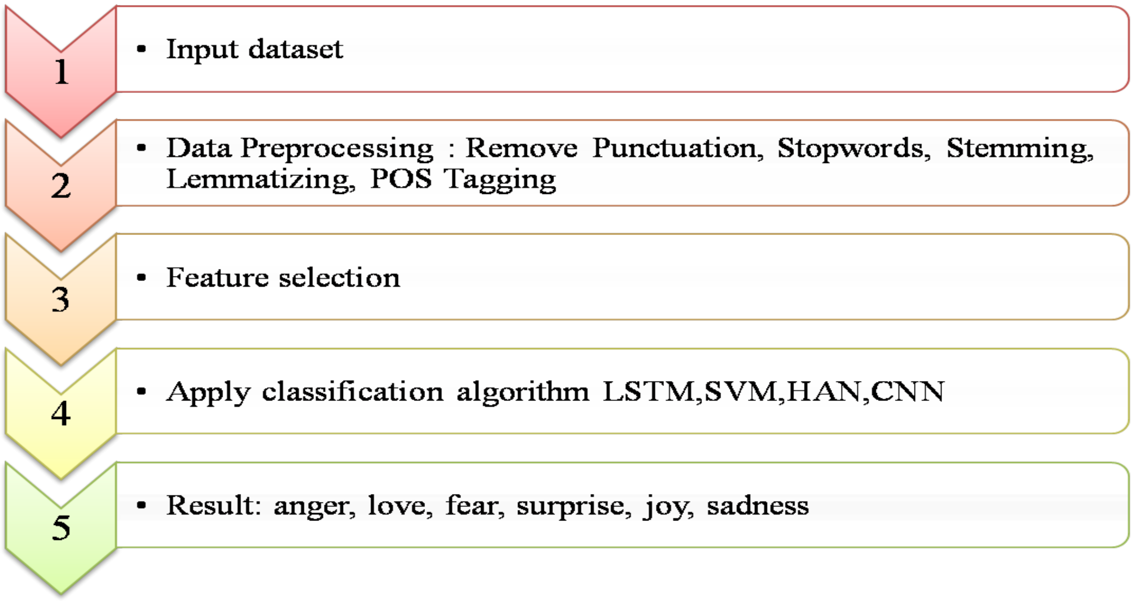


Figure 3.1: process flow

3.1 Implementation Process flow:

3.1.1 Step 1: Setting up the Development Environment

We will be using Spyder IDE that comes along with the anaconda installation to do all our programming.

3.1.2 Step 2: Choosing Your Dataset

We first need data. we can get textual data from any website like a movie review website, or Amazon product reviews, and so on. Here, we'll be using a labelled textual dataset. The dataset we are using has 416809 tweets in total, labelled into different human sentiments.We import the basic libraries and then read the dataset.

If we want to perform emotion analysis on real-time tweets we have connect our twitter API to python program.In order to use Twitter's API, we have to create a developer account on the Twitter apps site as per following steps.

1. Log in or make a Twitter account at https://developer.twitter.com

2. Create a new app.

3. Fill in the app creation page with a unique name, a website name, and a project description.

4. Once the application is created , open the 'Keys and Access Token' tab.

5. Copy 'Consumer Key', 'Consumer Secret', 'Access token' and 'Access Token Secret'.

6. Establish the connection with Twitter API.

3.1.3 Step 3: Preprocessing the Data

First, we have to bring some uniformity to the text by making everything lowercase, removing punctuation, and stop words (like prepositions), using tokenization, stemming , finding synonyms and antonyms lemmatizing and by speech tagging.

- Making all letters lowercase:

- Removing Punctuation, Symbols:

- Lemmatisation:

  To gain any proper insight, we need to get all the words to their root form, i.e the variants of a word within the text (for example plural forms, past tense, etc) must all be converted to the base word it represents. This is called lemmatisation. Along with that, we have added code to revert repetition of letters in a word with the assumption that hardly any word has letters repeated more than twice, consecutively. Though not very accurate, it can help in some corrections.

- Stemming

  stemming involves removing affixes from words and returning the root. Search engines like Google use this to efficiently index pages. The most common algorithm for stemming is the PorterStemmer. Let's take an example.
  stemmer.stem('writes')

‘write’

stemmer.stem(’writing’)

‘write’

stemmer.stem(’write’)

‘write’

- Finding antonyms and Synonyms:
  WordNet is an NLP database with synonyms, antonyms, and brief definitions. We have downloaded this with the NLTK downloader.

- Finding Stop words:
  We can filter NLTK stop words from text before processing it.
  example: from nltk.corpus import stopwords
  text=”Today is a great day. It is even better than yesterday. And yesterday was the best day ever!”
  output: [‘Today’, ‘great’, ‘day’, ‘.’, ‘It’, ‘even’, ‘better’, ‘yesterday’, ‘.’, ‘And’, ‘yesterday’, ‘best’, ‘day’, ‘ever’, ‘!’]

- Tokenizing Text:
  Before processing the text in NLTK Python Tutorial, you should tokenize it. What we mean is you should split it into smaller parts- paragraphs to sentences, sentences to words. We have two kinds of tokenizers- for sentences and for words.for example,

  *Sentence Tokenizer
  text=”Today is a great day. It is even better than yesterday. And yesterday was the best day ever.”
  from nltk.tokenize import sent_tokenize
  sent_tokenize(text)
  [‘Today is a great day.’, ‘It is even better than yesterday.’, ‘And yesterday was the best day ever.’]
  *Word Tokenizer
  nltk.word_tokenize(text)
  [‘Today’, ‘is’, ‘a’, ‘great’, ‘day’, ‘.’, ‘It’, ‘is’, ‘even’, ‘better’, ‘than’, ‘yesterday’, ‘.’,‘And’, ‘yesterday’, ‘was’, ‘the’, ‘best’, ‘day’, ‘ever’,‘.’]

- Speech Tagging
  NLTK can classify words as verbs, nouns, adjectives, and more into one of the following classes:

  1. CC coordinating conjunction
  2. CD cardinal digit
  3. DT determiner
  4. EX existential there
  5. FW foreign word
  6. IN preposition/subordinating conjunction
  7. JJ adjective ‘big’
  8. JJR adjective, comparative ‘bigger’
  9. JJS adjective, superlative ‘biggest’
  10. LS list marker 1)
  11. MD modal could, will
  12. NN noun, singular ‘desk’
  13. NNS noun plural ‘desks’
  14. NNP proper noun, singular ‘Harrison’
  15. NNPS proper noun, plural ‘Americans’
  16. PDT predeterminer ‘all the kids’
  17. POS possessive ending parent‘s
  18. PRP personal pronoun I, he, she
  19. PRP$ possessive pronoun my, his, hers
  20. RB adverb very, silently,
  21. RBR adverb, comparative better
  22. RBS adverb, superlative best
  23. RP particle give up

24. TO to go ‘to‘ the store.
25. UH interjection
26. VB verb, base form take
27. VBD verb, past tense took
28. VBG verb, gerund/present participle taking
29. VBN verb, past participle taken
30. VBP verb, sing. present, non-3d take
31. VBZ verb, 3rd person sing. present takes
32. WDT wh-determiner which
33. WP wh-pronoun who, what
34. WP$ possessive wh-pronoun whose
35. WRB wh-abverb where, when

For example, text=’I am a human being, capable of doing terrible things’
sentences=nltk.sent_tokenize(text)
for sent in sentences:
print(nltk.pos_tag(nltk.word_tokenize(sent)))
[(‘I’, ‘PRP’), (‘am’, ‘VBP’),(‘a’, ‘DT’), (‘human’, ‘JJ’), (‘being’, ‘VBG’), (‘,’, ‘,’),(‘capable’, ‘JJ’), (‘of’, ‘IN’), (‘doing’, ‘VBG’), (‘terrible’, ‘JJ’), (‘things’, ‘NNS’)]

3.1.4 Step 4: Feature Extraction

Once you make the text data clean, precise, and error-free, each sentence is represented by a group of keywords. Now, we need to perform ‘Feature Extraction’, i.e extracting some parameters from the data that can be presented numerically. We consider two different features, TF-IDF and Count Vectors.
Split the data into training and testing parts before performing feature extraction.

Term Frequency-Inverse Document Frequency (TF-IDF):
TF-IDF an acronym than stands for “Term Frequency – Inverse Document Frequency”, is used for Word vectorization i.e., general process of turning a collection of text documents into numerical feature vectors. TF-IDF are word frequency

scores that try to highlight words that are more interesting, e.g. frequent in a document but not across documents. This parameter gives the relative importance of a term in the data and is a measure of how frequently and rarely it appears in the text.

Term Frequency: This summarizes how often a given word appears within a document.

Inverse Document Frequency: This down scales words that appear a lot across documents.

$$TF(term) = \frac{number\_of\_times\_the\_term\_appears\_in\_document}{Total\_numbers\_of\_terms\_in\_document} \tag{3.1}$$

$$IDF(term) = \frac{total\_number\_of\_documents}{number\_of\_documents\_with\_term\_in\_it} \tag{3.2}$$

$$TF - IDF = TF(term\_in\_document) * IDF(term) \tag{3.3}$$

Count Vectors:

This is another feature we consider and as the name suggests we transform our tweet into an array having the count of appearances of each word in it. The intuition here is that the text that conveys similar emotions may have the same words repeated over and over again.

3.1.5 Step 5: Training Our Models

With the numerical representations of sentences ready, we can directly use them as inputs for some classic machine learning models. Here, we trained our learning model using 4 different algorithms such as Long Short-Term Memory (LSTM), Convolutional Neural Network(CNN), Support vector machine(SVM) and Hierarchical Attention Network (HAN) . These methods can, in fact, be used for tackling any kind of classification problem. In our case, we want to classify if a given tweet into one of the 6 emotions. For this we are using a dataset which contains 416809 tweets to train our model.

3.1.6 Step 6 :Testing

Now test how it performs in reality by giving this model some random text input.

Chapter 4

Result

4.1 Results for offline dataset

After training the models using 4 different algorithms we get output as follows which shows the iterations and time taken by model to perform each iteration. Also accuracy and loss at each iteration. The learning curves show how accuracy and loss varies from 1st iteration to Nth iteration. Accuracy is a measure of how accurate our model prediction is compared to true data and it is in the % format. Loss is a sum of error made in training or validating data.

After training dataset by Convolutional Neural Network(CNN) we got results as follows:

```
In [29]: history=model.fit(x_train, y_train, validation_data=(x_val, y_val),epochs=3, batch_size=5,callbacks=[cp])

Train on 375129 samples, validate on 41680 samples
Epoch 1/3
375129/375129 [==============================] - 14441s 38ms/step - loss: 0.7354 - acc: 0.7942 - val_loss: 0.7433 - val_acc: 0.
7944

Epoch 00001: val_acc improved from -inf to 0.79441, saving model to model_cnn.hdf5
Epoch 2/3
375129/375129 [==============================] - 42482s 113ms/step - loss: 0.9837 - acc: 0.7856 - val_loss: 1.0717 - val_acc:
0.7958

Epoch 00002: val_acc improved from 0.79441 to 0.79578, saving model to model_cnn.hdf5
Epoch 3/3
375129/375129 [==============================] - 15220s 41ms/step - loss: 1.1924 - acc: 0.7789 - val_loss: 1.1172 - val_acc: 0.
6983

Epoch 00003: val_acc did not improve from 0.79578
```

Figure 4.1: Iterations for CNN algorithm

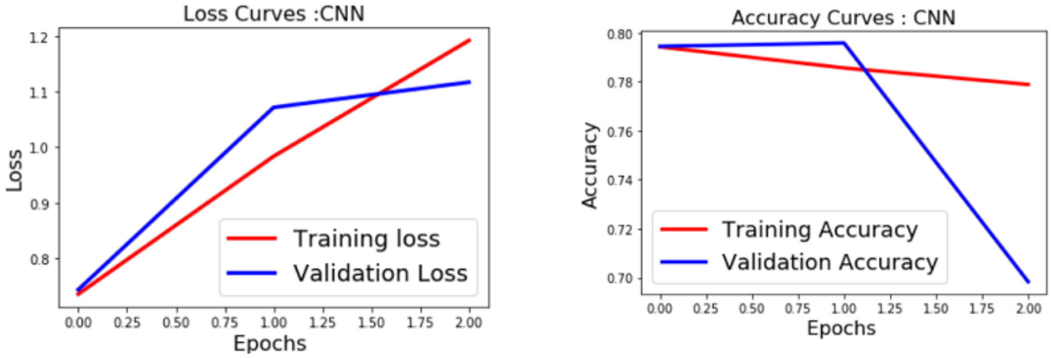


Figure 4.2: accuracy and loss curves for CNN

```
Train on 375128 samples, validate on 41681 samples
Epoch 1/10
375128/375128 [==============================] - 5548s 15ms/step - loss: 0.1962 - accuracy: 0.9045 - val_loss: 0.0935 - val_acc
uracy: 0.9399
Epoch 2/10
375128/375128 [==============================] - 5303s 14ms/step - loss: 0.0904 - accuracy: 0.9414 - val_loss: 0.0904 - val_acc
uracy: 0.9392
Epoch 3/10
375128/375128 [==============================] - 5398s 14ms/step - loss: 0.0858 - accuracy: 0.9432 - val_loss: 0.0918 - val_acc
uracy: 0.9397
Epoch 4/10
375128/375128 [==============================] - 5248s 14ms/step - loss: 0.0840 - accuracy: 0.9435 - val_loss: 0.0912 - val_acc
uracy: 0.9411
Epoch 5/10
375128/375128 [==============================] - 5443s 15ms/step - loss: 0.0811 - accuracy: 0.9440 - val_loss: 0.0919 - val_acc
uracy: 0.9410
Epoch 6/10
375128/375128 [==============================] - 5347s 14ms/step - loss: 0.0800 - accuracy: 0.9444 - val_loss: 0.0933 - val_acc
uracy: 0.9402
Epoch 7/10
375128/375128 [==============================] - 5294s 14ms/step - loss: 0.0790 - accuracy: 0.9443 - val_loss: 0.0954 - val_acc
uracy: 0.9402
Epoch 8/10
375128/375128 [==============================] - 5316s 14ms/step - loss: 0.0785 - accuracy: 0.9447 - val_loss: 0.1010 - val_acc
uracy: 0.9404
Epoch 9/10
375128/375128 [==============================] - 5861s 16ms/step - loss: 0.0778 - accuracy: 0.9447 - val_loss: 0.1022 - val_acc
uracy: 0.9403
Epoch 10/10
375128/375128 [==============================] - 37334s 100ms/step - loss: 0.0777 - accuracy: 0.9449 - val_loss: 0.1020 - val_a
ccuracy: 0.9408
```

Figure 4.3: Iterations after using LSTM for training purpose

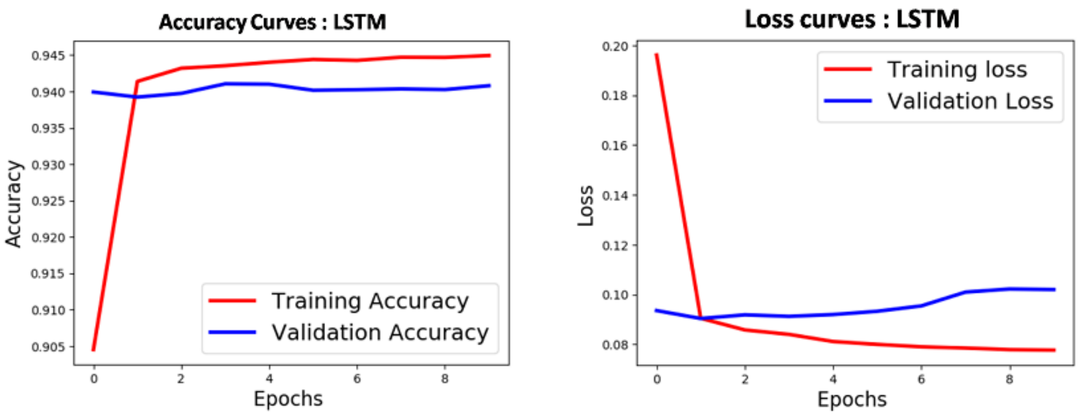


Figure 4.4: accuracy and loss curves for LSTM

```
In [18]: cp=ModelCheckpoint('model_han_.hdf5',monitor='val_acc',verbose=1,save_best_only=True)
         history=model.fit(x_train, y_train, validation_data=(x_val, y_val),
                   epochs=3, batch_size=5,callbacks=[cp])

         Train on 375129 samples, validate on 41680 samples
         Epoch 1/3
         375129/375129 [==============================] - 167525s 447ms/step - loss: 0.1615 - acc: 0.9240 - val_loss: 0.1171 - val_acc:
         0.9373

         Epoch 00001: val_acc improved from -inf to 0.93731, saving model to model_han_.hdf5
         Epoch 2/3
         375129/375129 [==============================] - 173336s 462ms/step - loss: 0.1160 - acc: 0.9381 - val_loss: 0.1108 - val_acc:
         0.9368

         Epoch 00002: val_acc did not improve from 0.93731
         Epoch 3/3
         375129/375129 [==============================] - 169453s 452ms/step - loss: 0.1195 - acc: 0.9389 - val_loss: 0.1381 - val_acc:
         0.9360

         Epoch 00003: val_acc did not improve from 0.93731
```

Figure 4.5: Iterations after using HAN algorithm

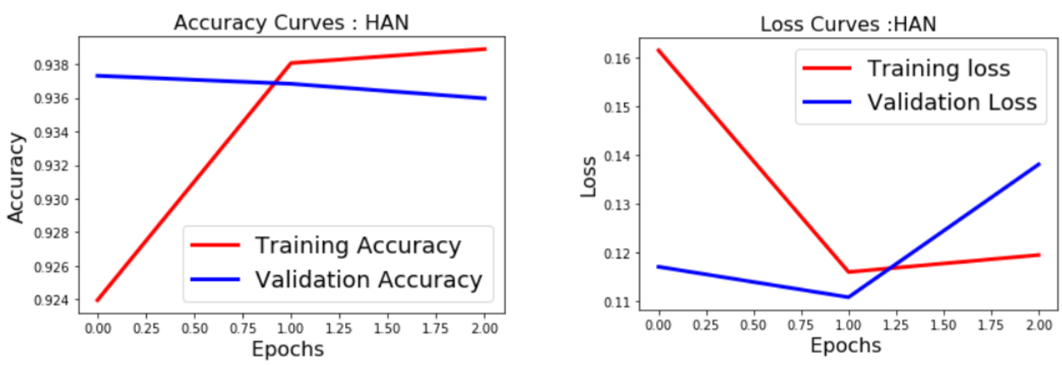


Figure 4.6: accuracy and loss curves for HAN algorithm

```
In [5]: runfile('C:/Users/NimishaJadav/Desktop/project/emotion_csv.py', wdir='C:/Users/
NimishaJadav/Desktop/project')
C:/Users/NimishaJadav/Desktop/project/emotion_csv.py:92: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/
user_guide/indexing.html#returning-a-view-versus-a-copy
  data.text[i] = words
              precision    recall  f1-score   support

       anger       0.87      0.89      0.88      5658
        fear       0.81      0.86      0.83      4809
         joy       0.91      0.91      0.91     14155
        love       0.76      0.72      0.74      3455
     sadness       0.93      0.92      0.92     12088
    surprise       0.71      0.68      0.70      1516

    accuracy                           0.88     41681
   macro avg       0.83      0.83      0.83     41681
weighted avg       0.88      0.88      0.88     41681

processing time: 17992.54545068741 seconds
```

Figure 4.7: Results after training model using SVM

Algorithm	Time/Epoch	Training Accuracy	Validation Accuracy
LSTM	9113s-10 epochs	94.56	94.11
HAN	159274s-1 epoch	93.56	93.80
SVM	23856s	88.00	-
CNN	24047s-3 epochs	77.89	69.83

Table 4.1: Summary of training and validating accuracy of 4 algorithms

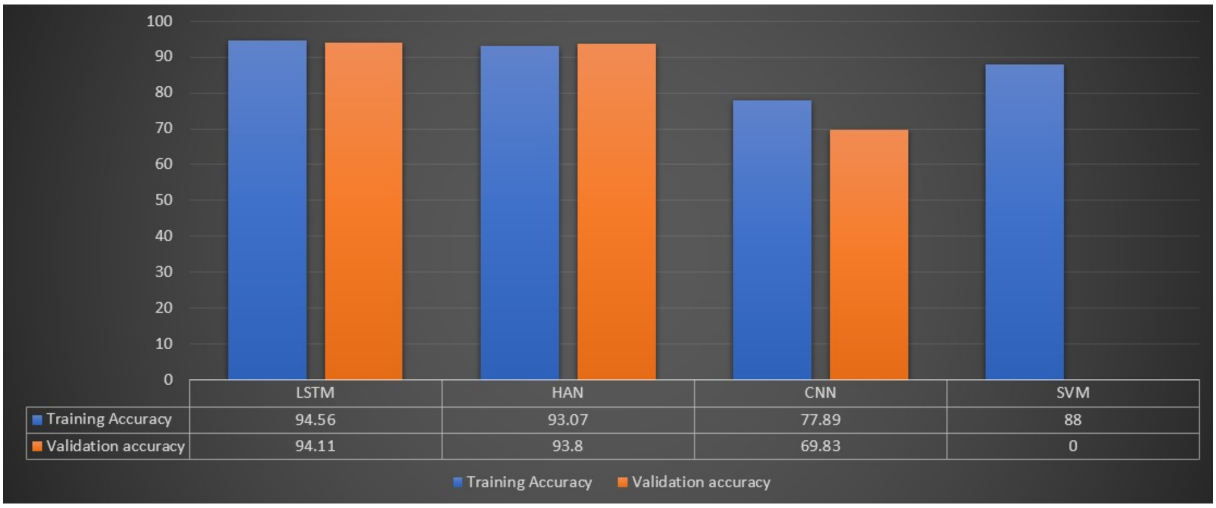


Figure 4.8: accuracy of 4 different algorithm

In figure 4.8 and table 4.1 we can see that LSTM algorithm gives output in less amount of time and it has highest validation and training accuracy. Out of all the 4 models LSTM gave the best outcomes.

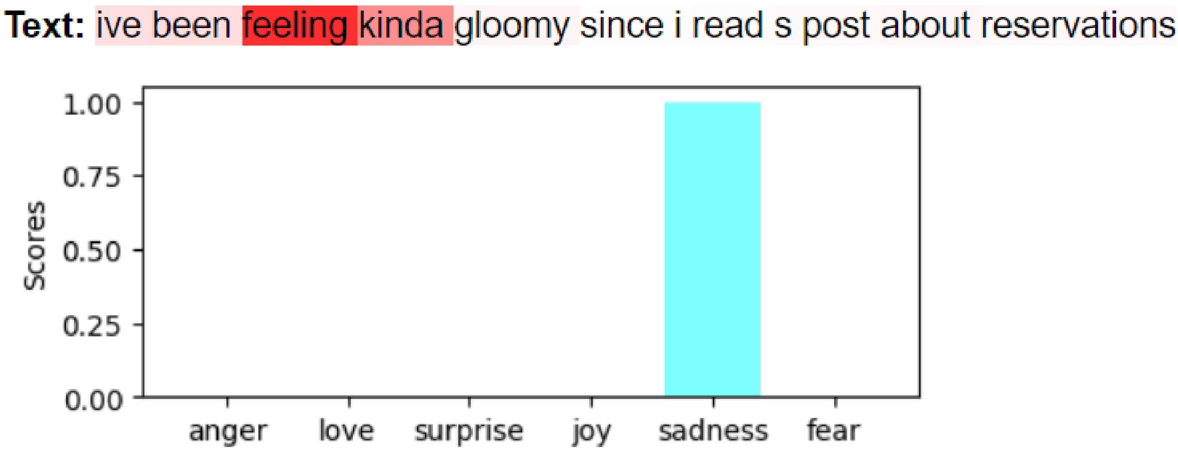


Figure 4.9: Extracting emotions from sentence

After training when we test our model using LSTM algorithm which showed best performance in training we got output as shown in figure 4.9.

4.2 Results for real-time data

The Twitter API platform offers two options for streaming realtime Tweets. Each option offers a varying number of filters and filtering capabilities.

We have performed emotion analysis for real time data as well, for different topics and we change the tweet count every time. For this real-time data we choose current most discussed topic on twitter like CAA bill, Coronavirus outbreak and Yesbank controversy.

No of tweets	CAA	Coronovirus	yesbankTime
5000	65.38% neutral	40.22% neutral	55.52% neutral
10000	62.88% neutral	44.77% neutral	53.84% neutral
20000	54.16% neutral	40.58% neutral	58.84% neutral
30000	53.85% neutral	39.71% neutral	60.06% neutral

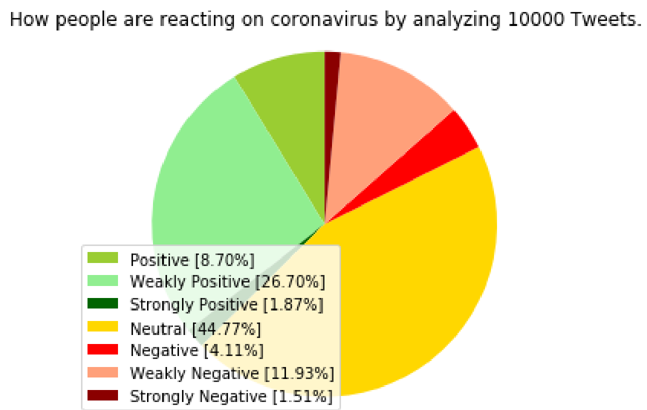
Table 4.2: Result for real-time data

```
Enter Keyword/Tag to search about: coronavirus

Enter how many tweets to search: 10000
How people are reacting on coronavirus by analyzing 10000 tweets.

General Report:
Weakly Positive

Detailed Report:
8.70% people thought it was positive
26.70% people thought it was weakly positive
1.87% people thought it was strongly positive
4.11% people thought it was negative
11.93% people thought it was weakly negative
1.51% people thought it was strongly negative
44.77% people thought it was neutral
```




```
Conclusion
44.77% neutral
```

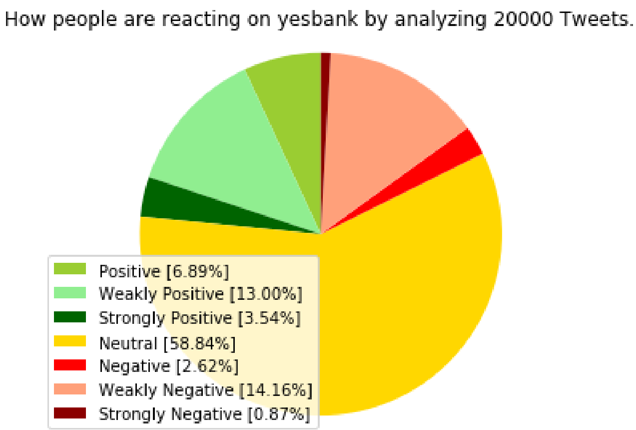
Figure 4.10: Emotion analysis for 10,000 tweets

```
Enter Keyword/Tag to search about: yesbank

Enter how many tweets to search: 20000
How people are reacting on yesbank by analyzing 20000 tweets.

General Report:
Weakly Positive

Detailed Report:
6.89% people thought it was positive
13.00% people thought it was weakly positive
3.54% people thought it was strongly positive
2.62% people thought it was negative
14.16% people thought it was weakly negative
0.87% people thought it was strongly negative
58.84% people thought it was neutral
```




```
Conclusion
58.84% neutral
```

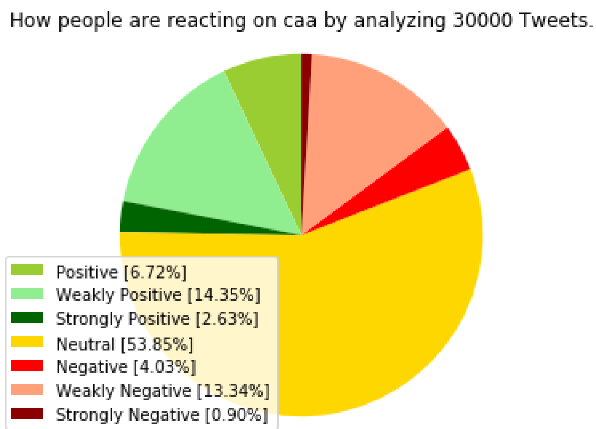
Figure 4.11: Emotion analysis for 20,000 tweets

```
Enter Keyword/Tag to search about: caa

Enter how many tweets to search: 30000
How people are reacting on caa by analyzing 30000 tweets.

General Report:
Weakly Negative

Detailed Report:
6.72% people thought it was positive
14.35% people thought it was weakly positive
2.63% people thought it was strongly positive
4.03% people thought it was negative
13.34% people thought it was weakly negative
0.90% people thought it was strongly negative
53.85% people thought it was neutral
```




```
Conclusion
53.85% neutral
```

Figure 4.12: Emotion analysis for 30,000 tweets

Chapter 5

Conclusion and future scope

In the era of the rapid development of social network and internet of things, it is very meaningful to explore the emotional recognition of user-generated data through technology. In our project, we have implemented emotion recognition method from text, based on the network which consists of CNN, LSTM, HAN and SVM Algorithm.

A novel model is implemented and applied on real-world datasets. In the experimental work, we have conducted the realtime data and offline datasets and then analyzed using the 4 classification machine learning and deep learning models. Experiments on emotionally related datasets shows that our method can achieve better performances compared with baseline methods. Our framework is general enough to be applied to more scenarios. Experiments proved that human emotion recognition is a challenge due to the subjectivity of language and phenomena such as irony or sarcasm.

In the future works, we will extend this emotion encoding to other tasks such as affective computing and sentiment analysis. It is possible to improve the performance of this model by using the larger emotion word embeddings.In future, this method can be develop to find emotion like skepticism, hope, anxiety, excitement.

Also same emotion recognition model can be implement for data which is in audio format.

Publication

Based on all the the experiments that we have done we have published two research papers in international journals.

1. V. Preethi, Nimisha Jadav, Komal Shirsat, Mohan Bonde, "Emotion Recognition from Text using LSTM algorithm," International Journal of Computer Sciences and Engineering, Vol.8, Issue.6, pp.30-34, 2020.

2. V. Preethi, Nimisha Jadav, Komal Shirsat, Mohan Bonde, "Emotion extraction from text: Analysis of four different algorithms", Journal of Emerging Technologies and Innovative Research (JETIR), Volume 7, Issue 6, June 2020.

IJCSE

International Journal of Computer Sciences and Engineering

Scholarly Peer-Reviewed Scientific Research Publishing Journal

Publication Certificate

This is to certify that

Nimisha Jadav

has published a paper entitled "Emotion Recognition from Text using LSTM algorithm" in International Journal of Computer Sciences and Engineering, Volume-8, Issue-6, Jun 2020, after review reports of our editorial board and review board.

We wish you for your success and bright future.......

Managing Director
IJCSE, ISSN: 2347-2693
www.ijcseonline.org
editor@ijcseonline.org

JCSE

International Journal of Computer Sciences and Engineering

Scholarly Peer-Reviewed Scientific Research Publishing Journal

Publication Certificate

This is to certify that

Komal Shirsat

has published a paper entitled "Emotion Recognition from Text using LSTM algorithm" in International Journal of Computer Sciences and Engineering, Volume-8, Issue-6, Jun 2020, after review reports of our editorial board and review board.

We wish you for your success and bright future.......

Managing Director
IJCSE, ISSN: 2347-2693
www.ijcseonline.org
editor@ijcseonline.org

International Journal of Computer Sciences and Engineering

Scholarly Peer-Reviewed Scientific Research Publishing Journal

Publication Certificate

This is to certify that

V. Preethi

has published a paper entitled "Emotion Recognition from Text using LSTM algorithm" in International Journal of Computer Sciences and Engineering, Volume-8, Issue-6, Jun 2020, after review reports of our editorial board and review board.

We wish you for your success and bright future.......

Managing Director

IJCSE, ISSN: 2347-2693

www.ijcseonline.org

editor@ijcseonline.org

International Journal of Computer Sciences and Engineering

Scholarly Peer-Reviewed Scientific Research Publishing Journal

Publication Certificate

This is to certify that

Mohan Bonde

has published a paper entitled "Emotion Recognition from Text using LSTM algorithm" in International Journal of Computer Sciences and Engineering, Volume-8, Issue-6, Jun 2020, after review reports of our editorial board and review board.

We wish you for your success and bright future.......

Managing Director
IJCSE, ISSN: 2347-2693
www.ijcseonline.org
editor@ijcseonline.org

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal

www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Nimisha Jadav

In recognition of the publication of the paper entitled

Emotion extraction from text: Analysis of four different algorithms

Published In JETIR (www.JETIR.org) ISSN UGC Approved (Journal No: 63975) & 5.87 Impact Factor

Published in Volume 7 Issue 6 , June-2020 | Date of Publication: 2020-06-22

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2006302 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2006302** **Registration ID : 234305**

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal
www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of
Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)
Is hereby awarding this certificate to

Komal Shirsat

In recognition of the publication of the paper entitled

Emotion extraction from text: Analysis of four different algorithms

Published In JETIR (www.JETIR.org) ISSN UGC Approved (Journal No: 63975) & 5.87 Impact Factor

Published in Volume 7 Issue 6 , June-2020 | Date of Publication: 2020-06-22

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162
Journal of Emerging Technologies and Innovative Research

JETIR2006302 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2006302** **Registration ID : 234305**

Journal of Emerging Technologies and Innovative Research
An International Open Access Journal
www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

V.Preethi

In recognition of the publication of the paper entitled

Emotion extraction from text: Analysis of four different algorithms

Published In JETIR (www.JETIR.org) ISSN UGC Approved (Journal No: 63975) & 5.87 Impact Factor

Published in Volume 7 Issue 6 , June-2020 | Date of Publication: 2020-06-22

ISSN 2349-5162

EDITOR

EDITOR IN CHIEF

JETIR2006302 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2006302** **Registration ID : 234305**

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal

www.jetir.org | editor@jetir.org

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Prof. Mohan Bonde

In recognition of the publication of the paper entitled

Emotion extraction from text: Analysis of four different algorithms

Published In JETIR (www.JETIR.org) ISSN UGC Approved (Journal No: 63975) & 5.87 Impact Factor

Published in Volume 7 Issue 6 , June-2020 | Date of Publication: 2020-06-22

ISSN 2349-5162

EDITOR

EDITOR IN CHIEF

JETIR2006302 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2006302** **Registration ID : 234305**

References

[1] Chetan R. Chopade. Text based emotion recognition: A survey. *International Journal of Science and Research (IJSR)*, 2013.

[2] M. Li E. Batbaatar and K. H. Ryu. Semantic-emotion neural network for emotion recognition from text. *IEEE Access*, 2019.

[3] F. Lanubile F. Calefato and N. Novielli. In *EmoTxt: A toolkit for emotion recognition from text*, 2017.

[4] Mahira Kirmani Mudasir mohd Nida Manzoor Hakak, Mohsin Mohd. *Emotion Analysis: A Survey.* International Conference on Computer, Communications and Electronics (Comptelix)Manipal University Jaipur, Malaviya National Institute of Technology Jaipur and IRISWORLD, 2017.

[5] Dr. Ashok Kumar R Priyanka H S, Ramya B V. Classification model to determine the polarity of movie review using logistic regression. *International Research Journal of Computer Science (IRJCS)*, 2019.

[6] R. Zatarain-Cabada R. Oramas-Bustillos, M. L. Barron-Estrada and S. L. Ramírez-Ávila. A corpus for sentiment analysis and emotion recognition for a learning environment. 2018.

[7] H. Hajj S. Shaheen, W. El-Hajj and S. Elbassuoni. Emotion recognition from text based on automatically generated rules. *IEEE International Conference on Data Mining Workshop, 2014*, 2014.

[8] S. N. Shivhare and S. K. Saritha. Emotion detection from text documents. In *International Journal of Data Mining and Knowledge Management Process*, 2014.

ACKNOWLEDGEMENT

We have a great pleasure to express our gratitude to all those who have contributed and motivated during our project work. For selecting a topic of our choice and studying and understanding it,is a good opportunity to learn and to know new technology ,understand the way project are researched and executed. We would like to thank our guide Prof. Mohan Bonde, our HOD Dr.Sanjay Shitole and to all the teachers who help made us understand our topic well.

We would like to thank the institution Usha Mittal Institute of Technology, for giving us the opportunity to present a Major-Project. Last but certainly not the least, we would like to thank our families for their wholehearted support and overall guidance in assisting us to accomplish this undertaking.

Date:

Nimisha Jadav
Komal Shirsat
V.Preethi