ASPECT BASED SENTIMENTAL ANALYSIS USING GATED RECURRENT UNIT

A PROJECT REPORT

Submitted by

ASHA GUTLAPALLI [Reg No: RA1611003010352]
SHRAVANI DIXIT [Reg No: RA1611003010492]

Under the guidance of

Mrs. SINDHU C

(Assistant Professor, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

S.R.M. Nagar, Kattankulathur, Kancheepuram District

MAY 2020

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**ASPECT BASED SENTIMENTAL ANALYSIS USING GATED RECURRENT UNIT**" is the bonafide work of " **ASHA GUTLAPALLI [Reg No: RA1611003010352], SHRAVANI DIXIT [Reg No: RA1611003010492]**", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs. SINDHU C
GUIDE
Assistant Professor
Dept. of Computer Science & Engineering

SIGNATURE

Dr. B. AMUTHA
HEAD OF THE DEPARTMENT
Dept. of Computer Science & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

As social creatures, we depend on each other to make decisions and form opinions. The reviews reachable are not only useful for helping customers to choose a product or service, but also the manufacturers and industrialists to strategize their production and marketing tactics as well as government organizations to listen to its people. Various social media platform embedded with recommender systems, market researches and predictions can be made more practical and comprehensive by sentiment analysis. But when the need to find elaborate details squeezed into concise sentences, aspect-level sentiment analysis is the field to turn to. The advancements in deep learning approaches has widened horizons in various applications, including those Aspect-based Sentiment Analysis. Their major advantages include computational strength and the ease with which they are trained. Numerous Natural Language Processing techniques have been cast into the neural models to make up for the semantic complications which these networks can't handle by themselves. Scientists have been inventive with creating novel models, merging many of neural networks and attention mechanisms to elevate the performances in aspect detection and sentiment polarity identification on diverse datasets, like SemEval, Amazon reviews, and Twitter. This field is quite promising as its independent nature compels researchers to probe further into the topic. In this paper, we strived to integrate Gated Recurrent Unit, which takes advantage of parallel gating mechanism giving high performance speed. The dense layers improve the accuracy by learning from the previous layers while embedding the data which follows tokenisation, class encryption and arrangement. The proposed model has shown remarkable results, with much better accuracy in both aspect categorization and sentiment classification on the dataset of SemEval 2015.

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to the Vice-Chancellor, Dr. Sandeep Sancheti, for the opportunity of representing our esteemed institute as well as opening the doors to a brighter future.

We thank our Director, Dr. C. Muthamizhchelvan, for the knowledge and experiences that we have received which will inspire us to be better engineers and professionals for the rest of our lives.

We convey our hearty gratefulness to the Head of the Computer Science and Engineering Department, Dr. B Amutha, for providing us with an excellent atmosphere for doing research and demonstrating our skills.

We gladly thank our class in-charges, Mrs. P Akilandeswari and Mrs. S Nagadevi for their efforts in our inspiration and management of our time and skills in our projects as well as academic schedules.

We would include our Project Panel Head, Mr. Mathew Eliazer, for the constant monitoring of our project and the encouragement our efforts in creating something beneficial for the people.

Last but not the least, we thank our guide, Mrs. Sindhu C, for the relentless support of our endeavours, advising along each step we strode, suggesting measures and improving our design and our efficiency. In spite of her busy schedule, she invested her time in guiding us to the right direction.

ASHA GUTLAPALLI

SHRAVANI DIXIT

TABLE OF CONTENTS

ABSTRACT **iii**

ACKNOWLEDGEMENT **iv**

LIST OF TABLES **viii**

LIST OF FIGURES **xi**

LIST OF ABBREVIATIONS **xii**

LIST OF SYMBOLS **xiii**

1 INTRODUCTION **1**

2 LITERATURE REVIEW **7**

2.1 Literature Review Tabulation of Related Work 7

2.2 Analysis from Aspect Based Sentimental Analysis Research Papers 13

3 SYSTEM DESIGN **14**

3.1 Data Collection 15

3.1.1 Online Datasets 15

3.2 Data Pre-Conditioning 16

3.2.1 Missing Data 16

3.2.2 Different Data Formats 16

3.2.3 Noisy Data 17

3.2.4 Inaccurate Data 17

3.3 Text Pre-conditioning 18

3.4 Text Cryptography 19

3.5 Text Formatting 19

3.6 Augmentation 20

3.7 Attribute Association 20
3.8 Gated Recurrent Unit 21
3.8.1 Deep Learning 22
3.8.2 Gated Recurrent Unit Algorithm 22
3.8.3 Aspect Based Sentimental Analysis 24
3.9 Learning 25
3.10 Prediction 26

4 SYSTEM IMPLEMENTATION 28
4.1 Programming language 28
4.1.1 Python 28
4.2 Integrated Development Environment 29
4.2.1 Jupyter Notebook 29
4.3 Graphical User Interface 29
4.3.1 Anaconda Navigator 30
4.4 Library Packages 30
4.4.1 Keras and Tensorflow 30
4.4.2 Pandas 31
4.4.3 Numpy 31
4.4.4 Matplotlib 32
4.5 Dataset 33
4.6 Data pre-treatment 34
4.6.1 Tokenization 34
4.6.2 Word Filing 34
4.6.3 Class encryption 35
4.6.4 Arrangement 36
4.6.5 Filling 36
4.7 Facet Establishment 37
4.7.1 Token Embedding 37
4.8 Bi-Directional Gated Recurrent Unit 38
4.8.1 Activation Function 39
4.8.2 Loss Calculator 40

4.8.3 Optimizer 41
4.9 Training 41
4.10 Testing 42

5 RESULT AND DISCUSSION **44**

6 CONCLUSION **50**

REFERENCES **51**

APPENDIX A **52**

APPENDIX B **58**

LIST OF TABLES

2.2 Literature Review Tabulation of Related Work 12

5.1 Testing 44

LIST OF FIGURES

1.1 The various techniques used in Sentiment Analysis apart from neural networks . . . 3

1.2 Challenges faced by the domain of sentiment analysis . . . 4

1.3 Why is Sentiment Analysis important . . . 6

3.1 System Design . . . 14

3.2 Online Datasets . . . 15

3.3 Missing Data . . . 16

3.4 Different Data Formats . . . 17

3.5 Noisy Data . . . 17

3.6 Inaccurate Data . . . 18

3.7 Text Pre-conditioning . . . 18

3.8 Text Cryptography . . . 19

3.9 Text Formatting . . . 20

3.10 Augmentation . . . 20

3.11 Attribute Associations . . . 21

3.12 Deep Learning . . . 22

3.13 Algorithm . . . 23

3.14 Aspect Establishment . . . 24

3.15 Sentiment Classification . . . 25

3.16 Learning . . . 26

3.17 Prediction . . . 27

4.1 Python . . . 28

4.2 Jupyter Notebook . . . 29

4.3 Anaconda Navigator . . . 30

4.4 Keras and Tensorflow . . . 31

4.5 Pandas . . . 31

4.6 Numpy 32
4.7 Matplotlib 32
4.8 System Implementation 33
4.9 Tokenization 34
4.10 Word Filing 35
4.11 Class encryption 35
4.12 Arrangement 36
4.13 Filling 37
4.14 Token Embedding 38
4.15 Bi-Directional Gated Recurrent Unit 38
4.16 Rectified Linear Units 39
4.17 Softmax 40
4.18 Sparse Categorical Entropy 40
4.19 Adam 41
4.20 Training 42
4.21 Testing 42

5.1 Performance of Aspect Categorization 44
5.2 Performance for Sentiment Classification 45
5.3 Cost of Aspect Categorization 46
5.4 Cost of Sentimental Analysis 46
5.5 Embedding Projection for Aspect Categorization 47
5.6 Embedding Project for Keyboard Aspect 47
5.7 Embedding Projection for Aspect Categorization 48
5.8 Embedding Project for Like Sentiment 49

9.1 Dataset Output 58
9.2 Tokenization and Word Filing Output 59
9.3 Arrangement Output 59
9.4 Filling Output 60
9.5 Class Encryption Output 60
9.6 Model for Aspect Establishment Output 60

9.7 Model for Sentiment Classification Output 61
9.8 Accuracy and Output for Model 1 61
9.9 Accuracy and Output for Model 2 Part 1 62
9.10 Accuracy and Output for Model 2 Part 2 62
9.11 Output for User Instance Part 1 63
9.12 Output for User Instance Part 2 63

LIST OF ABBREVIATIONS

ABSA	Aspect Based Sentimental Analysis
Bi-LSTM	Bidirectional Long Short Term Memory
CNN	Convolutional Neural Network
CRF	Conditional Random Field
GRU	Gated Recurrent Unit
GSD	Google Similarity Distance
GUI	Grpahical User Interface
IDE	Integrated Development Environment
LSTM	Long Short Term Memory
NGD	Normalised Google Distance
PSO	Particle Swarm Optimization
PSRTN	Position and Self-attention mechanism R-Transformer Network
ReLU	Rectified Linear Units
RMSprop	Root Mean Square Propagation
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent

LIST OF SYMBOLS

b_c	Cell Memory Bias Vector
W_c	Cell Memory Weight Vector
$C^{<\text{t}>}$	Current Cell Memory Vector
$\otimes$	Element-wise Multiplication
$X^{<t>}$	Input Vector
$\hat{C}^{<t>}$	New Cell Memory Vector
$C^{<t-1>}$	Previous Cell Memory Vector
b_r	Reset Gate Bias Vector
τ_r	Reset Gate Vector
W_r	Reset Gate Weight Vector
σ	Sigmoid Function
$tanh$	Tanh Function
b_u	Update Gate Bias Vector
τ_u	Update Gate Vector
W_u	Update Gate Weight Vector

CHAPTER 1

INTRODUCTION

We live in the modern era of online shopping and E-commerce with enormous amount of data generated being generated on a routine basis. This information invites the challenge to be segregated and labelled for real-world interpretation and understanding. Researchers and scientists are regularly inventing, modifying and innovating existing technology to deal with the ever-increasing need for the categorization of the produced data. The resulting techniques, mechanisms and algorithms that have been invented for Aspect-Based Sentiment Analysis have shown to have applications in other fields like speech recognition, sarcasm detection etc and vice versa. Sentiment Analysis, the superset of Aspect Based Sentimental Analysis, is the to-go chant to deal with this mammoth task.

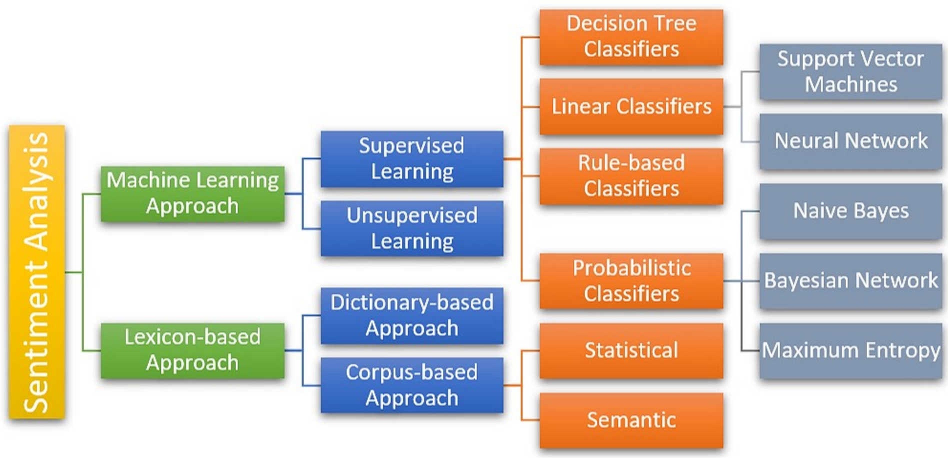
Sentiment Analysis is nothing but the very interpretation of the reviews that any service has received and classifying them into groups and categories for easy navigation and construal. The customers can thus navigate these categories to search for what they are looking. This makes the task for the consumer much more convenient as they do not have to go through and break down each and every review for a product or a service they wish to purchase. Instead they can look up the aspects they desire and find the features, opinions, pros and cons in a remarkably easier fashion and make a self-suited decision. Sentiment Analysis deals with polarity and identifies them as three basic categories- positive, negative and neutral. The level of this analysis is myriad, ranging from a single sentence, involving a whole paragraph, or a whole document. Customers have been more vocal and opinionated than ever before. They have received a platform where they can discuss practically anything, a dish-washer or a politician. We can find pointers, analysis and summarisations of almost every product, brand or service at one click. Businesses depend severely on customer feedback through social media platforms, consumer surveys and their personal feedback centres to make amends and improvisations on their investments and strategies.

There are many types of Sentimental Analysis as mentioned. Fine-grained sentiment analysis includes two more categories in addition to the basic three to perceive a stronger inclination or its lack thereof in a company's asset. Emotion detection is as the name suggest, recognizing the emotion of the public towards the resource. This branch is a lot more complicated because of the complications of human speech and vernacular, which changes on a daily basis and could mean multiple things with a single reference. Multilingual sentimental analysis is intricate as it incorporates many languages which themselves involve a burdensome of pre-processing and resources. The last of Sentiment Analysis is Aspect-Based Sentiment Analysis.

ABSA has garnered the most attention because of the crisp attention to detail that accompanies it. Earlier studies in this field counted in many machine learning techniques, supervised and unsupervised, considering manual attempts as irrational [2]. But with the rise of deep learning approaches, which have their own applications in Artificial Intelligence, scholars have included neural networks into the process. This is due to the computational powers of such networks which enhances the performance of the model because of parallel computing and gating mechanisms. This reduces the convergence time. However, recurrent and convolutional neural networks don't quite turn up to the expectations as they don't show relations among the target terms and don't take the full text into account [4]. Convolutional models also ignore separate modelling. However, the addition of linguistic hints such as position of words, Part-of-speech taggers, domain-specific word embedding has proven to improve polarity indicators.

The combination of neural networks credited with Natural Language techniques has shown promising results. Long short-term memory and attention mechanism models require quite a bit of training. Context-dependent sentiment features have been put into use in recurrent neural networks. Various other novel architectures like Graph based Convolutional models, transformer-based memory networks, bilinear dependency tress are built to deal with ABSA using neural networks [1], [9], [11] as in Figure 1.1.

Figure 1.1: The various techniques used in Sentiment Analysis apart from neural networks



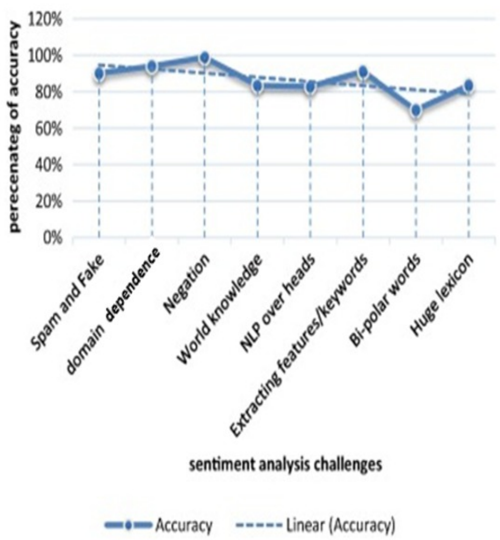
Aspect-based sentiment analysis has attracted more interest and attention possibly due to the fact it is the more difficult part of Sentiment Analysis, as it depends on parts of sentences. It is mentioned in [8] how ignoring Aspect extraction, the most important subtask in ABSA, profoundly reduces the accuracy of the classification. Aspect-based Categorization indicates Sentiment Polarity and Aspect-term Sentiment Analysis drains out explicit terms [6].

Neural Networks can however ignore the effect of text structure in target aspects [9] and suffer from low performance due to the semantic complexity of the natural languages [7]. But by inventive structures that include innovation like Bilinear Dependency Tree [6] or Graph Convolutional Model [9], in which the latter takes advancement in treating text like graph which boosts the application in computational models. By indulging in creating awareness of word orders [10] or shift words which direct the sentiment polarity, like “but”, “however”, [6] the performance of the neural networks receives a major enhancement.

In layman’s terms the challenges in ABSA are plenty. The scientists have to reckon with simpler aspects like subjectivity and tone. The inaccuracy in the predictions of these aspects have proven to be of disastrous consequences as the wrong interpretation can change the meaning of a sentence entirely. Subjective and objective texts as well the tone of them need to correctly recognized. Failing to find the difference between the two has impacted the conclusions of the mechanisms deployed. The context of the text is just as vital. They might have a reference to the previous products or rival brands which equally influence the resultant label. The context can change the polarity of the sentences and an appreciative utterance might be translated by the machine as a negative comment. In the techniques at our disposal currently, more often than not, these context terms are supposed to fed into the algorithm explicitly.

Many people use irony and satire to express themselves in day-to-day life. These figures of speeches have paved their way into the way they do so in the virtual world as well. However, machines are not capable of combating this form of communication just yet. That doesn't stop the people from passing on a sarcastic comment on their purchase. There aren't textual cues for machines to learn or even doubt in the slightest the satirical indentation of a sentence which categorize sentences with positive attributes under negative reviews since they mean the exact opposite of what has been stated. Comparisons are also vital in tackling with Sentimental Analysis, proving the importance of context as in Figure 1.2.

Figure 1.2: Challenges faced by the domain of sentiment analysis



Sentiment Analysis Challenges	Percentage of Accuracy	Percentage of Linear Accuracy
Spam and Fake	90	96
Domain Dependence	96	95
Negation	99	92
World Knowledge	81	90
NLP over heads	80	85
Extracting features/ keywords	93	82
Bi-polar words	67	80
Huge lexicon	83	78

The object of comparison needs to be found precisely along with the comparison direction which throws light on what basis the comment has been constructed. Emojis have in popular culture made their way into the way people communicate with each other, and depicting their sentiments. These emojis could be shorter sequences of letters, consisting of two letters mostly, or longer sequences of usually a vertical nature.

We need to attend to an in-depth character and word level while analysing these reviews and statements. The pre-processing of content, which might classify the features and distinguish them to truly and improve the precision. Lastly, the definition of neutral is another challenge to accurately execute sentiment analysis. The tagging of data has to be consistent which will lend a helping hand to the model in training while it tries to decipher the definition of neutral for future purposes. It has been already displayed that the inaccuracy in finding a neutral component drastically impacts the accuracy of the model. The various types of neutral comments also need to be paid heed to as there might be a sneaky subjective review which was complicated for the machine to decode.

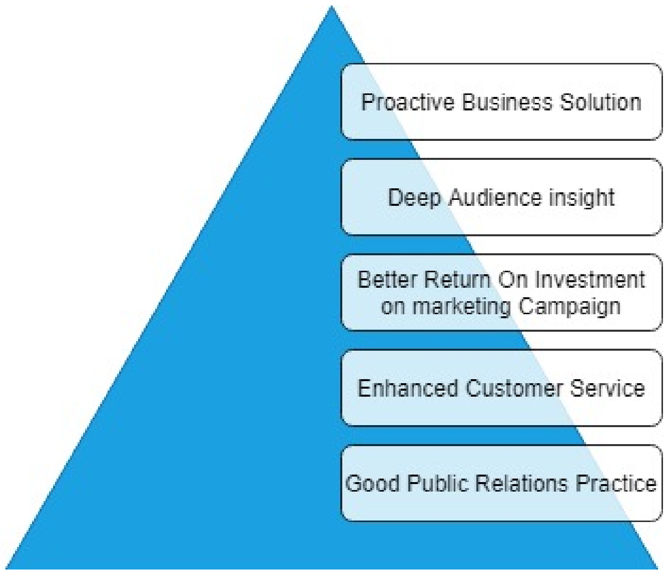
The primary reason why there is so much research happening in the field of sentiment analysis and so much labour is being taken on the shoulders is because the long-term utilities are immense. Humans themselves have difficulty in pin-pointing the emotions and aspects of an utterance. So it is not just the demerits of the technology we possess, but a short-coming in our own cognitive capabilities. The inclusion of sentiment analysis in a business has seen improvements and developments of the company as there can be a better atmosphere to create and distribute. The economic benefits are plenty as the monetary aspects are now wisely placed. The resource of time is also productively spent as the demands and requirements of the customers are swiftly and efficiently dealt with.

Social media is the modern backbone of people's lives in this era. People rely heavily on what's being said about whom through various social media platforms and forums. The nightmares of the customers and delights of a consumer are accessible on the tip of one's fingers, literally. The scope and reach of these platforms are vast and thus absolutely beneficial, or conversely disastrous, for the public image of a product, service, brand or a company. Sentiment analysis is useful in dialling down a magnified response and can be swerved to the more essential topics for discussion.

The applications of Sentiment analysis are various. That is illustrated by the following Figure 1.3.

The noise, or the chatter on the online platforms is too much to deal with when you only want to reckon with meaningful and deep insights. The quality of conversations can be filtered through the mechanisms as well create an interactive and responsive

Figure 1.3: Why is Sentiment Analysis important ?



feedback station. This helps us to track trends and prioritize actions in the benefits of both the producer and the consumer.

Apart from social media, there are other dimensions on the internet which people access to validate and consolidate their opinions and thoughts. Most of the telecommunications industry is now available on internet platforms and can be viewed by the people if and when they want. They can track specific events happening around the globe and read related articles of the same as they have been separated for their convenience. Online team members are assigned to cover these events as per their expertise and availability, thus highlighting once again, that sentiment analysis is just as beneficial to the employers as to the clients. Lastly, customer feedback is not only used to improve the products and services, but the surveys themselves are notched up a quality which give a wider insight of the audience's minds. Customers can clear their doubts and ask their questions and receive suggestions and answers in an automated fashion without much effort. Market researches are profited as they can be used to study the competition and further enhance their assets and increase customer satisfaction.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review of Aspect Based Sentimental Analysis using Deep Learning

The following table Table 2.1 captures the literature of all the papers we surveyed.

The primary criteria in conducting the following literature survey was the recency of the topic. To ensure that we devised a unique and improved model, we have extensively surveyed any model that resembles our field of interest and application. They have used certain techniques in a sub-step of their model different from the proposed model, and have themselves conducted the experiments with and without the addition of the main mechanism and altercations of their models, which have varied the results in each performance.

Neural networks have been evolved in its usage and application over the years, as observed. The tweaks and twists that have been rendered into the classic models is worth noticing. The transfer from robust machine learning algorithm, through innovative inclusion of fuzzy logic to neural networks shows the shift of the area of studies of the scholars and researchers. More and more methods and techniques are designed and fused into the existing ones to bring forth the best possible result with immaculate precision and performance.

Paper No.	Method/ Technique used	Pros	Cons	Dataset	Accuracy
[1]	Interactive Gated Convolutional Network	Mutual correlation between target and context	Performance drops after 5 epochs	SemEval 2014 (laptop and restaurants)	81.34 % (restaurants), 75.24 % (laptop) [1]

continued on next page

continued from previous page

[2]	Co-occurrence matrix	Easy to train, outperforms supervised methods	Parameter setting	SemEval-2014	67.0 % (F1) [2]
[3]	Sequential rule-based approach with Google Similarity Distance (GSD), Particle Swarm Optimization (PSO) and Normalised Google Distance (NGD)	Novel, implementation in real-life reviews, diverse domains	Limited to explicit aspects, lexicon-dependent	-	- [3]
[4]	Recurrent Neural Network (RNN), Convolutional Neural Network (CNN)	Improvement in sentiment detection	Lowered accuracy in aspect category detection	SemEval-2016 (restaurant reviews)	89.21 % (Convolutional), 89.53 % (Long Short Term Memory (LSTM)) [4]

continued on next page

continued from previous page

[5]	Fuzzy Logic	Decrease in neutral values	Not Specified	-	- [5]
[6]	LSTM, dependency tree, Conditional Random Field (CRF)	Novel, applicable in other labelling tasks	Full layer connection absent	SemEval-2014 (restaurant reviews)	84.83 % (F1) [6]
[7]	LSTM, Markov Logic Network, SenHint	Aspect polarity not required	Dependent on training data	SemEval 2015 and 2016	81.96 % (2015 restaurant) and 86.87 % (2016 laptop) [7]
[8]	LSTM, SenticNet	High Accuracy	Parameter setting	Restaurant review data in Yangon	87.2 % [8]
[9]	CNN, adjacency matrix	High Accuracy	Dependent on training data	SemEval 2014 and Twitter	78.12 % (restaurant), 70.53 % (laptop), 70.66 % (twitter) [9]

continued on next page

continued from previous page

[10]	LSTM, Gaussian Kernel	High performance in explicit polarity context	Cannot detect multiple aspects accurately, dependent on sequence and sentence length	SemEval 2014	72.8 % (laptop), 79.4 % (restaurants) [10]
[11]	Transformer-based network	Extracts long-term features	Performance reduction by repeated operation	Weibo, SemEval 2014	81.87% (Restaurant), 61.67% (Weibo) [11]
[12]	RNN (Gated Recurrent Unit (GRU)), multiple hops attention mechanism	Identifies implicit polarity	Wrong predictions for multiple negation, complex and neutral sentences	SemEval 2015	74.28% (F1) [12]

continued on next page

continued from previous page

[13]	Feature Enhanced Attention CNN-Bidirectional Long Short Term Memory (Bi-LSTM)	extract a higher-level phrase representation sequence	cannot obtain target information accurately	SemEval 2014	83.21% [13]
[14]	Bi-LSTM and CRF	Aspect extraction is improved	Not Specified	SemEval 2014	85.69% (Restaurants) and 80.13% (Laptop) (F1) [14]
[15]	Position and Self-attention mechanism R-Transformer Network (PSRTN) model	Gets better contextual semantic information	Efficiency can be improved	SemEval2014 and Twitter	83.8% , 80.9% , and 75.1% respectively [15]
[16]	Bi-LSTM layer, self-attention layer and softmax layer	Sentiment polarity classification accuracy improved	Semantic information needs better accuracy	SemEval 2014	79.14% [16]

continued on next page

continued from previous page

[17]	Soft Attention-based Bidirectional Long Short-Term Memory	Highest classification accuracy	New vocabulary can't be detected	Twitter	97.87% [17]
[18]	Convolutional Long Short-Term Memory	Assigns relative importance to time series data	Performance hindered by imbalanced training data	Amazon	88.82% (F1) [18]
[19]	Pre-trained deep self-attention encoder	Task formulation is avoided	Neutral and limited reviews are predicted incorrectly	SemEval 2014	84.7% (restaurant), 78.7% (laptop) [19]
[20]	Multi-head attention mechanism	Implementation parallel computing of sequence Elements, less noise	Needs improvements for external knowledge	SemEval 2014 and SemEval 2016	76.02% (laptop '14), 81.34% (restaurant '14), 88.00% (restaurant '16) [20]

Table 2.2: Literature Review of Aspect Based Sentimental Analysis

2.2 Analysis from Aspect Based Sentimental Analysis using Deep Learning

In the above table, it can be seen that training datasets including restaurants' reviews are easier to train since they have higher number of common categories- namely food and service. That is why, the comparative accuracy is higher. Neural networks by themselves lack high precision and thus need supporting mechanisms like attention mechanisms, Natural Language Processing Techniques and as seen above, innovative architecture. The datasets in these papers have mostly been benchmark datasets of SemEval 2014 to 2016 as well as real time datasets from Amazon, Twitter and local restaurants which directly apply their techniques in the real world.

Among the mentioned papers, the Soft Attention-based Bidirectional Long Short-Term Memory [17] model achieves the highest classification accuracy even though it lags in detecting new words. RNN and CNN, since achieving better precision than up-to-the-minute methods based on Machine Learning Algorithms even without any modifications in their structure, can be seen in [4]. Most of the models need rectifications to be used in the real-time systems as they have their performances reduced once new vocabulary is introduced. They have also been performing up to the expectations because they are heavily dependent on the training data [7],[9].

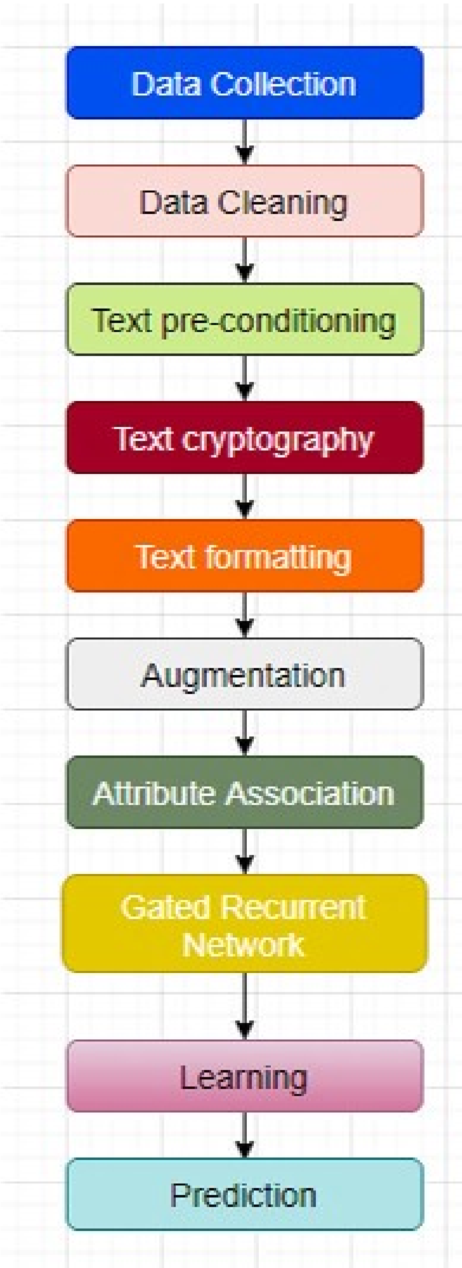
Neural networks also need parameter settings, like unsupervised methods, but their high accuracy is a major advancement from their predecessor [8], [9]. Moreover, these structures also need enhancements in classifying implicit aspects as well as neutral sentiments, both of which affect the performance of the model drastically. Many of them also struggle while dealing with complex sentences and multiple negations. As neural networks mainly fall behind in understanding semantic structures and nuances of the reviews, efforts have been made specifically in this area. In conclusion, these structures have many advantages over the existing methods, and have remarkable performance results in certain fields of Aspect-Based Sentiment Analysis.

CHAPTER 3

SYSTEM DESIGN

The system is designed in a module by module fashion, encapsulating a lot more stages in each of them. All these are expanded in the following sections starting from how the input acquaints to the anticipated output emergence as in Figure 3.1.

Figure 3.1: System Design



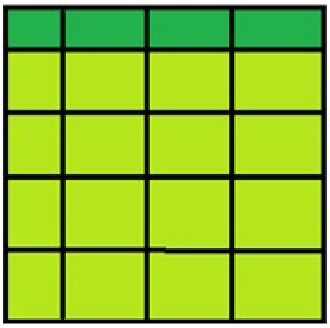
3.1 Data Collection

Data collection is the process of collecting data from various places or one place depending on the preference of the subject and engineer. It lays out the foundation of the whole system. It has the potential to very much make or break the system. The quality and the quantity of the data extracted must be the top priority of the individual responsible. There are many forms of data like structured, unstructured and, semi-structured determined by the source. Few likely techniques are elaborated.

3.1.1 Online Datasets

If there is not enough time for the two techniques mentioned above as it entails collection, cleaning, normalizing, formatting, people can fall back on online datasets for support. These online datasets are made by anyone literally a student or it might be a standard dataset made by professionals and used in multiple research papers as seen in Figure 3.2.

Figure 3.2: Online Datasets



Whichever way one decides to choose their dataset, they have to be wary about the amount of data they might call for even in the distant future. Furthermore, be cautious of the authenticity of the data and verify that it suits their purpose.

3.2 Data Pre-Conditioning

Data Preconditioning is a vital phase for the dataset before moving on to other phases. There multiple possible issues that can found in data. Some of the data cleaning procedures are discussed below.

3.2.1 Missing Data

Null and empty entries are the most faced concerns while analyzing the dataset. In that case, filling them with the average value can be or deleting those rows in another way of approaching it. But in doing so, there is a danger in losing essential information as seen in Figure 3.3. Hence that may not be recommended.

Figure 3.3: Missing Data

3.2.2 Different Data Formats

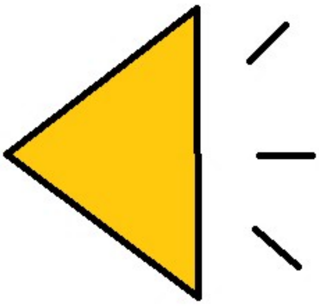
Since data is collected from various sources, there is a high chance of data organized in multiple ways as seen in Figure 3.4. To understand it easily, we need to make sure it all has a particular format. A great method of achieving this is data normalization. Data normalization averages out or evens out the plane of data.

Figure 3.4: Different Data Formats

3.2.3 Noisy Data

Noisy data is the unnecessary and unrelated facts in the data as seen in Figure 3.5. Those are tainted and deformed figures that either require to be weeded out or checked. The solutions to this problem are sorting and binning them into buckets and also gathering more statistics. This ensures reduce in noise.

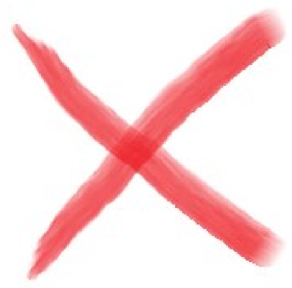
Figure 3.5: Noisy Data



3.2.4 Inaccurate Data

Incorrect data might be entered due to human error or even machine error as seen in Figure 3.6. It is time-consuming to correct each record manually. Before fixing the issue, we have to examine the dataset for possible faults. Then we can update the records using a function that will change them accordingly.

Figure 3.6: Inaccurate Data

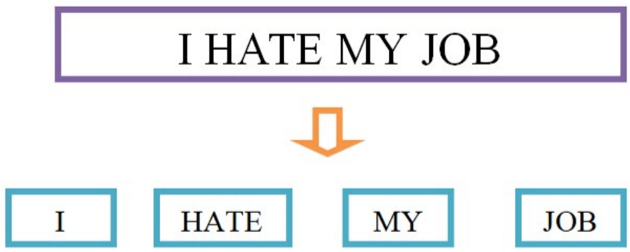


After completing these steps, we move on to the rather significant segments. Data has to be altered uniformly like converting it to lowercase, stemming the words, removing punctuation, and tokenization. This minimizes the redundancy. It also saves space and effort of dissection.

3.3 Text Pre-conditioning

Text Pre-conditioning is a method to condition the text before the algorithm is run on it. An important reason for doing this is to spoon-feed the code a simpler form of the original. Big parts are torn apart in smaller parts. This ensures that the program takes a closer look at the fundamental elements of the data. It is extremely difficult for it otherwise to digest the content and will seem complicated. Thereby, for grasping the knowledge, it is necessary to take a closer look at every constituent as in Figure 3.7.

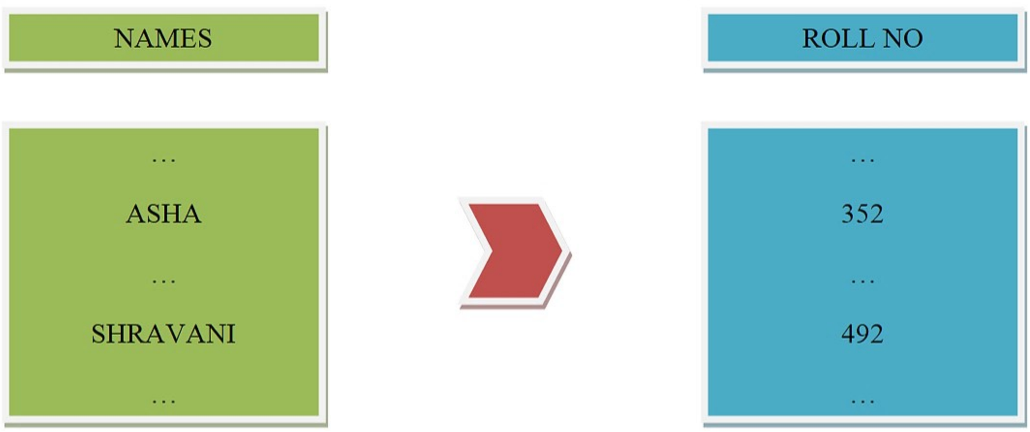
Figure 3.7: Text Pre-conditioning



3.4 Text Cryptography [23]

Text cryptography is a method to compress a long word into a single digit. It is sometimes a word or a few words in the case of text. Cryptography is used in various other fields for multiple reasons. But one of the most important reasons is to make identification easier. For example, in schools, students are allotted roll numbers and in colleges, undergraduates are allocated registration numbers. There is a pool of people; it is tricky to remember all their names as in Figure 3.8.

Figure 3.8: Text Cryptography



3.5 Text Formatting

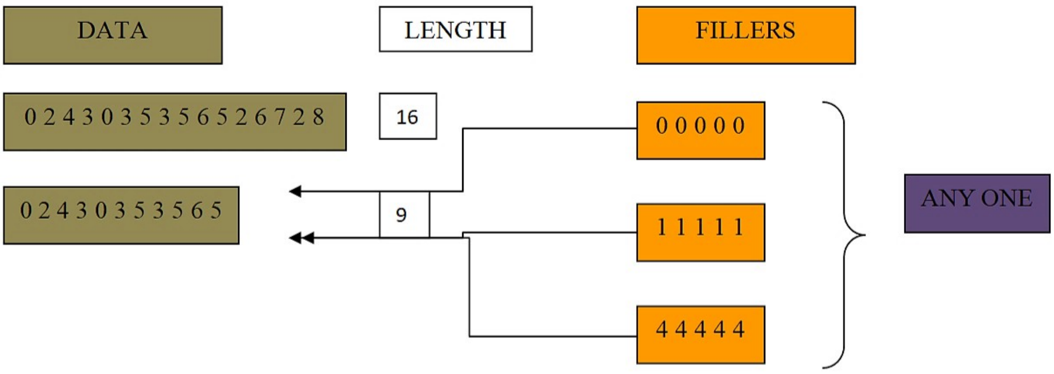
Text formatting is a method of providing a structure to the data. The data cannot be of different shapes, and sizes. They all must be uniform. If there are not uniform, an engineer has to spend the effort to write code to take care of each type of information. That is just costly and unnecessary. That is why people opt to put it through a step where it comes out all being the same. It can be thought of as to be a molder for clay. There are general rules and regulations to be followed as in Figure 3.9.

Figure 3.9: Text Formatting

3.6 Augmentation

Augmentation is a method where it cushioning something to make it the same size as the rest. Here, the addition of material or in this case data is done if the sentences are of dissimilar lengths to convert it into a standard length. Now, some will doubt the necessity of this. They may also be skeptical of the purpose of this. It is has a purpose when it comes to deep learning. In each layer, there are several neurons. To set a single number for neurons, it is imperative to do this as in Figure 3.10.

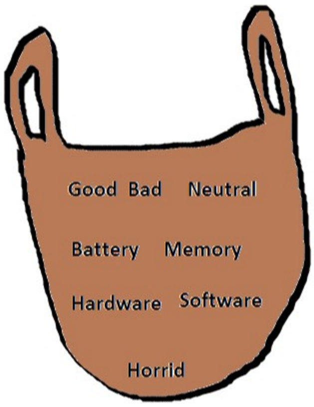
Figure 3.10: Augmentation



3.7 Attribute Association

Attribute Mining is the phase that may make or break the system. It is the process by which fundamental features are extracted for the model to assess. If the features are not appropriate then that will show in the model's ability to do well. Usually, there is a separate module for taking care of this task. But in some frameworks, the model itself will perform both attribute mining and running the actual algorithm upon the features drawn out. Some of the categories of attribute mining are explained as in Figure 3.11.

Figure 3.11: Attribute Associations



Associations are a type of attribute mining where there is a bag of features that are accumulated by calculating their distance from each other. If their relationship is closer then that particular attribute will be added to the structure and if they are sparsely related then that feature will be discarded. Some examples of this category are a bag of words, word embedding, and many more as seen in Figure 3.11.

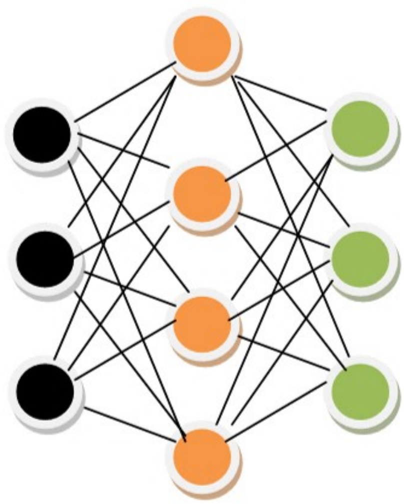
3.8 Gated Recurrent Unit

Paradigms are undoubtedly the most significant part of any framework. They comprise of the algorithm that extrapolates patterns from data. The input is sent to the paradigm after processing and it is expected to give the expected output. There are categorical data, numerical data, and also textual data. Machine Learning and Deep Learning are the two pillars of Data Science and Artificial Intelligence. Their algorithms can be solving a classification or clustering problem. There are indeed numerous algorithms capable of attaining great success with these practical applications but they are all grouped and presented under.

3.8.1 Deep Learning

Deep learning is a scrutinized area inside machine learning. Its popularity is due to its advanced trait organization by setting the weights and the bias as seen in Figure 3.12. They are made up of layers of neurons that compute the activations. There are three kinds of layers namely input, hidden and, output layers. Examples of this manner flexible with text processing are RNN, GRU and, LSTM.

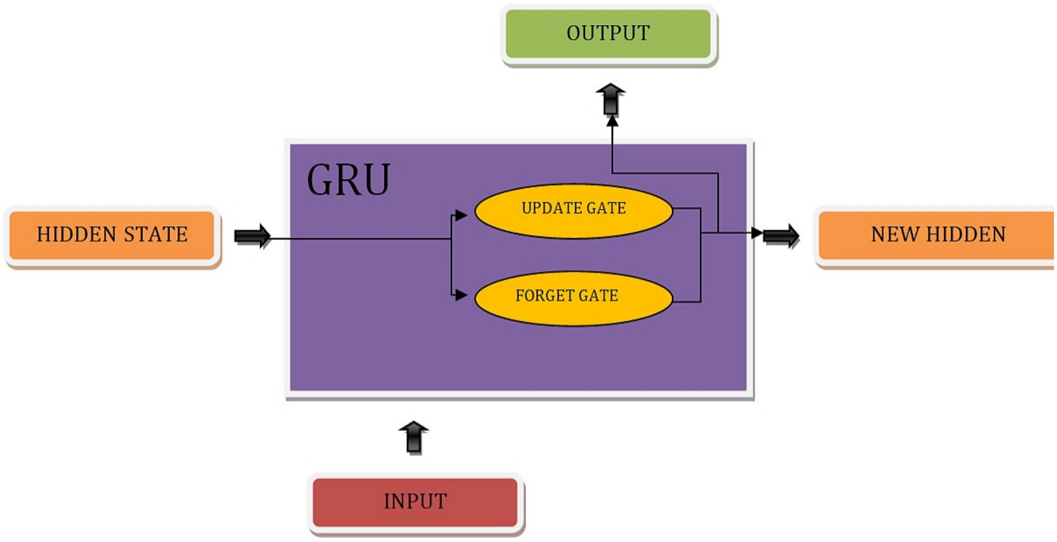
Figure 3.12: Deep Learning



3.8.2 Gated Recurrent Unit Algorithm

The algorithm implemented for our experiment is the GRU. [19] The GRU is a gated recurrent neural network. It is a deep learning mechanism. As mentioned before, it is a deep neural network. In our code, we implemented a deep and bi-directional GRU instead of only a forward one as seen in Figure 3.13. [20] It has few gates which are not logical like some may think. On the contrary, they are statistical gates. They aid in gauging activations with the help of a function.

Figure 3.13: Gated Recurrent Unit Algorithm



3.4.2.1 Update Gate

Update gate is a gate that determines how much of the old information must be brought into a new computation. The weights and biases are updated with accordance to Previous Cell Memory in a Sigmoid function and stored in a new vector. If we disregard this dependency, then it becomes hard to maintain continuity with syntax and semantics.

3.4.2.2 Forget Gate

Forget gate is the second gate that determines how much of the old information must be overlooked into a new computation. The weights and biases are altered with accordance to Previous Cell Memory in a Sigmoid function like the above. It is a consequence of this gate but its primary motive is not that. Its primary motive is to remove data that must be included while computation.

$$
\begin{aligned}
\Gamma_u &= \sigma \left(W_u \left[C^{<t-1>}, X^{<t>} \right] + b_u \right) \\
\Gamma_r &= \sigma \left(W_r \left[C^{<t-1>}, X^{<t>} \right] + b_r \right)
\end{aligned} \tag{4.4.1}
$$

The New Cell Memory is obtained by resetting the Previous Cell Memory with the help of weights and biases on input data in a Tanh function. Then the Previous and New Cell Memare updated. This latest cell memory is treated as the Current Cell Memory.

$$\hat{C}^{<t>} = tanh\,(W_c\,[\tau_r \otimes C^{<t-1>}, X^{<t>}] + b_c)$$
$$C^{<t>} = \tau_u \otimes \hat{C}^{<t>} + (1 - \tau_u\,) \otimes C^{<t-1>}$$

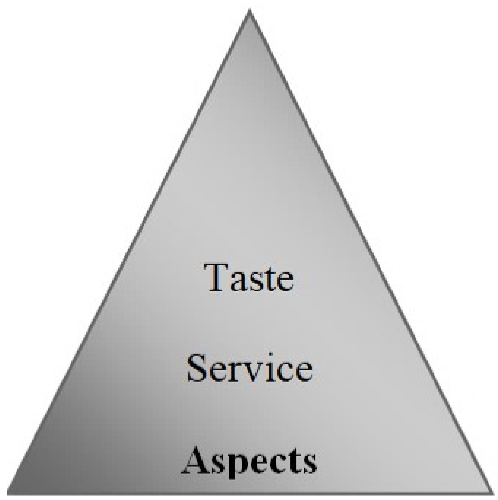
3.8.3 Aspect Based Sentimental Analysis

Since we are in quest of building an Aspect Based Sentimental Analysis framework, there would be two modules which are Aspect Establishment and, Sentiment Classification. Another optional module would be Aspect clustering if the existing framework is unable to handle so many trait classes. Aspect clustering as the name suggests executes clustering on aspects by compressing say 100 divisions into 20 divisions.

3.4.3.1 Aspect Establishment

Aspect establishment is a module where the properties of the data that will be significant for the paradigm are focused upon as seen in Figure 3.14. Without identifying these aspects, it is hard and impossible to jump to sentimental analysis. These aspects are nouns from the perspective of parts of speech. For instance, the aspect identification in the domain twitter analysis of politics will be accountability.

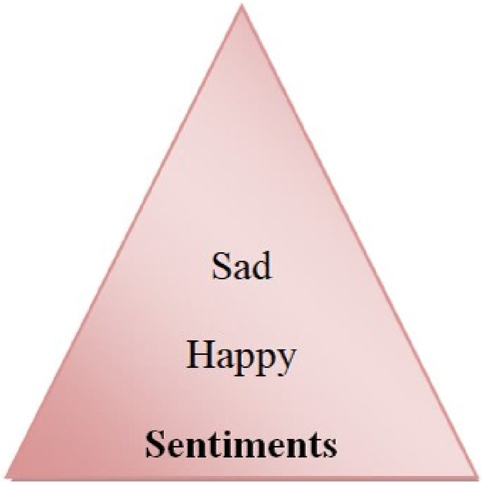
Figure 3.14: Aspect Establishment



3.4.3.2 Sentiment Classification

Sentiment classification is a module where the emotions of the data that were recognized previously are focused upon as seen in Figure 3.15. Broadly speaking, there are numerous emotions like positive, negative and, neutral. When it's a sentiment, nothing is black and white.

Figure 3.15: Sentiment Classification

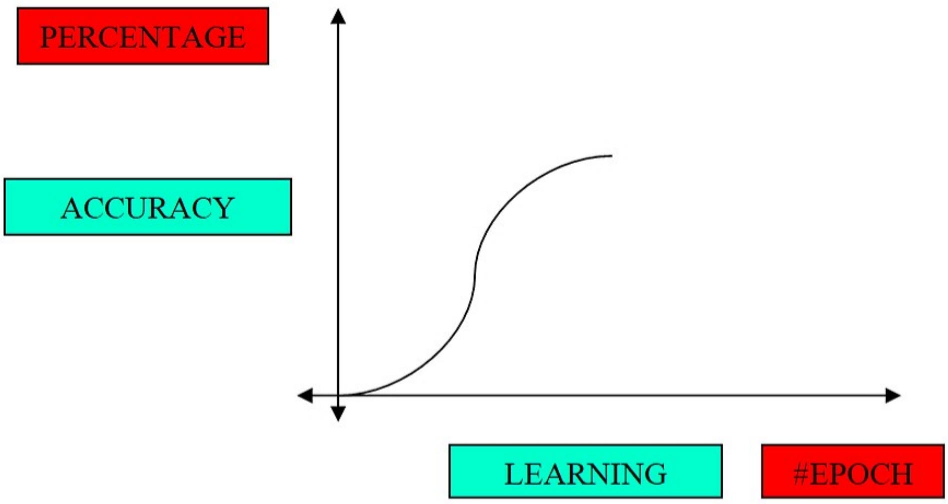


There is a scale of intensity that cannot be ignored. For instance, taking the same domain as before, the comments can say that the politician is accountable to the people. That is a positive sentiment.There are several approaches to this. We can integrate these two modules into one and have one algorithm take care of this or utilize the same algorithm separately on both tasks or implement different algorithms for these tasks. This is all up to the programmer.

3.9 Learning

For learning, the fit function is applied. This will let the input and output to be plotted. Now it is up to the program to sort out how the input and the output are related.

Figure 3.16: Learning

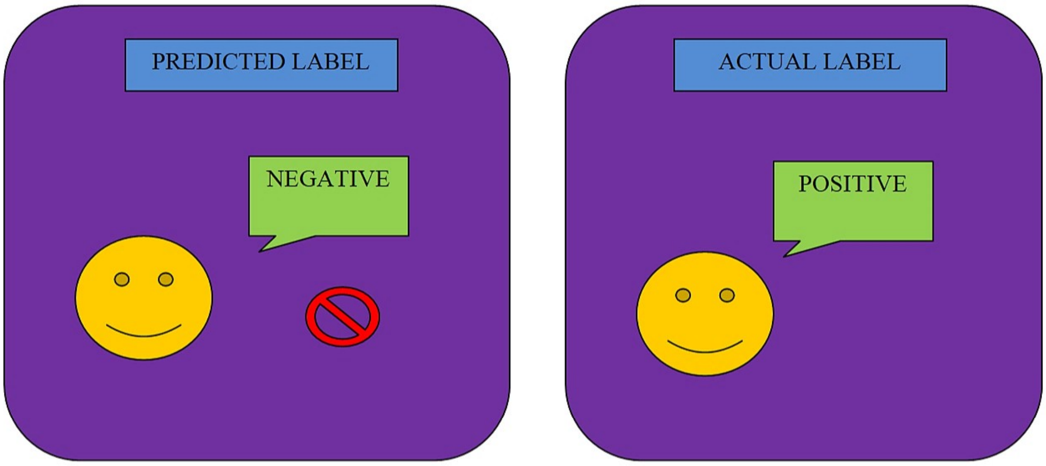


It picks the features whose weights and biases are more than the rest. The number of epochs will be set. Epoch is the number of times the batches will be trained. This is like a student preparing for an examination. They read once, then study it, and then revise their portion as in Figure 3.16. The machine performs an analogous task.

3.10 Prediction

For prediction, validation is carried out. Validation is a term that is quoted on many occasions. The student has prepared for the examination. Now they are facing the actual examination. It is to the teacher to see how well the student has performed and accordingly marks are given. The student sees the mark and assesses where they have gone wrong. They make sure that they do not make the same mistake again and get higher marks the next time. The code also undergoes a comparable prediction like this as in Figure 3.17.

Figure 3.17: Prediction



The theories highlighted in this chapter will eventually be put into practice in the next chapter. Then people look at a bigger picture of these ideas when they see them in action.

CHAPTER 4

SYSTEM IMPLEMENTATION

The hardware, software, tools and other set ups required before starting the implementation process are explained below one by one.

4.1 Programming language

To build any application, a programming language is a medium to communicate the programmer's objectives to the machine. It how they both chat. The programmer gives instructions and the machine implements the instruction.

4.1.1 Python

Python is the programming language of choice. Python, as many are aware, is the most user-friendly language. It is enriched with in-built functions. These functions prolong pages and pages of code but in python, it is just a few lines of code. Anyone who is not too aware of programming languages is also capable of using python for their apps. For data mining, python is the most widely employed programming language as in Figure 4.1.

Figure 4.1: Python

4.2 Integrated Development Environment

Integrated Development Environment is a platform where programming duties can be done. For each domain, there are respective Integrated Development Environment (IDE) platforms. All the sub-tasks like file handling, output window, variable handling, error window, visualization window, and source code can be viewed in parallel.

4.2.1 Jupyter Notebook

Jupyter Notebook is the IDE platform where coding took place for this experiment. Jupyter Notebook is popular for Machine Learning programming. In Jupyter Notebook, files can be uploaded from local file locations or create new files. It can manage many file types. The source code is split into cells where each cell can be run separately irrespective of the other surrounded code. There are special cells for special intentions as in Figure 4.2.

Figure 4.2: Jupyter Notebook

4.3 Graphical User Interface

Graphical User Interface is a user interface by which the customer can converse with the system. There are icons, pictures, audio and, video clips which translates a message that shows what it indicates. The user will click on the appropriate icon to do a job.

4.3.1 Anaconda Navigator

Anaconda Navigator is the GUI used where Jupyter Notebook resides. Anaconda Navigator has a lot more IDE platforms other than Jupyter Notebook like spyder, rstudio, orange3 and, many more. It has packages like Keras, matplotlib, and pandas which will be discussed further in detail. Different working environments can be customized like the default base root. Projects can be developed. There are learning chances on this Grpahical User Interface (GUI). People can reach out to other developers and experts in their communities as in Figure 4.3.

Figure 4.3: Anaconda Navigator

4.4 Library Packages

The library is like a library in real life where books are discovered to read. Likewise, in a library, some utilities execute a role. The library packages used in the experiment are explained as follows in the upcoming sections.

4.4.1 Keras and Tensorflow

Keras is a library that is used for deep learning and neural networks. All the algorithms in computer vision, natural language processing, speech processing can be accessed through this library. This how the GRU algorithm was instigated for this work. The model's layers like fully connected layers, embedding layers were put in operation with the help of Keras library as in Figure 4.4.

Tensorflow is a library package but also an open-source platform to code. It has

Graphical Processing Units which run faster. These can be contacted remotely. Keras is the library inside the Tensorflow library. Tensorflow is blowing up in terms of gaining people's and scientist's attention with its robustness.

Figure 4.4: Keras and Tensorflow

4.4.2 Pandas

Pandas is an admired library package. For importing and reading the dataset, Pandas was employed. Once the dataset is read or imported, the data structure is a Dataframe. Pandas is good at managing data structures and influencing its substance. Finally, when the operations are done, the final result can be again saved on the personal computer as in Figure 4.5.

Figure 4.5: Pandas

4.4.3 Numpy

When one hears the name of this package, they think of arrays. Numpy is indeed for an array, matrix, and other multidimensional data structures. The experiment called out for array operations for storing the data. Thus, Numpy library was the perfect library for that. Numpy is well-suited form Numerical procedures as in Figure 4.6.

Figure 4.6: Numpy

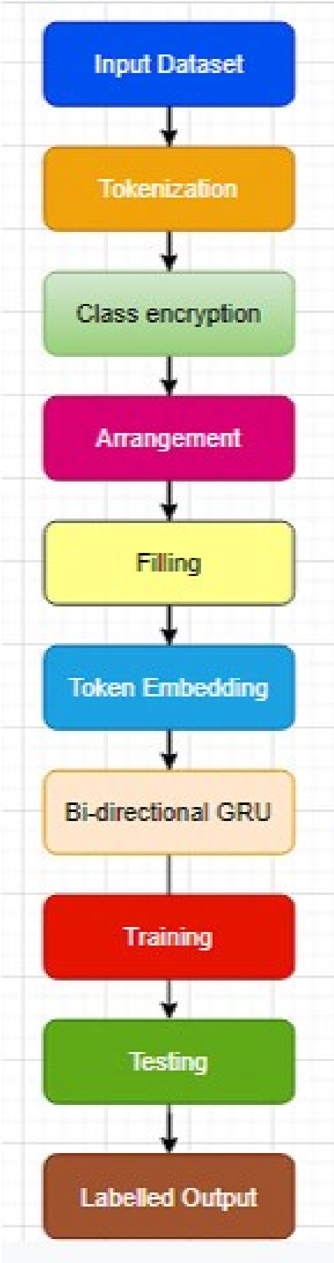
4.4.4 Matplotlib

Matplotlib is certainly the go-to library for visualization techniques. The experiment plotted two graphs for both Aspect Categorization and Sentiment Classification of the Train and Test accuracies from the start to the end of epochs. Line graphs were put in place to accomplish the above. There are much more attractive and complicated graphs too as in Figure 4.7.

Figure 4.7: Matplotlib

In system implementation, the nuances of the experiment are elaborated in detail. This will give people are clearer window to assimilate the theories in the previous chapter. All the particular options chosen throughout are specified as follows as in 4.8.

Figure 4.8: System Implementation



4.5 Dataset

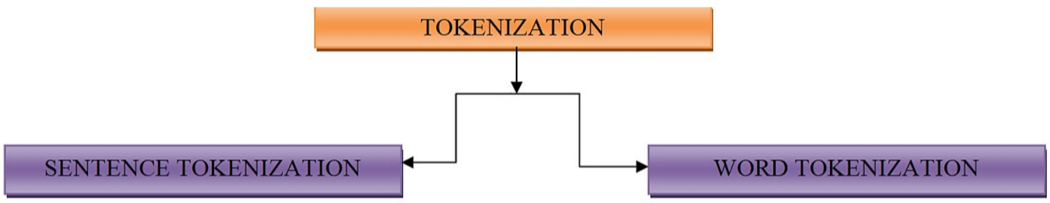
The dataset chosen for our code is the SEMEVAL 2015 Task 12. This is a popularly employed dataset for Aspect Based Sentimental Analysis. The breakdown of the dataset would be Sentence ID, Sentences, Aspect, and Polarity. This has annotated all the aspects and their respective sentiments. This has 3000 entries on record. Though like any dataset, it has few data cleaning issues too. We pondered upon web scraping data from online Electronic Commerce websites but finally though it would be a better idea to use a dataset that has been used by other researchers. This enables us to compare the performance of our results to that of others. Refer for the system design architecture.

4.6 Data pre-treatment

Data Pre-treatment consists of Tokenization, Word Filing, Class Encryption, Arrangement, and Filling as elaborated further below.

4.6.1 Tokenization

Figure 4.9: Tokenization

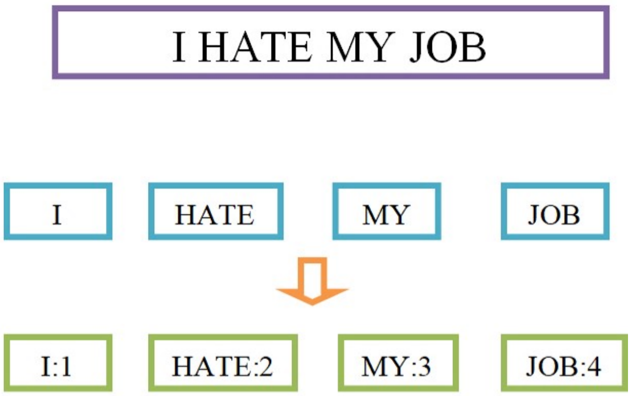


For any corpus, we first divide a huge entity into smaller entities so that they can be treated individually as seen in Figure 4.9. Tokenization also does a similar task but upon sentences in a text. First, the text is broken down into sentences and that is further broken down into words. Tokenization is the first step and before going to other pre-treatment steps we must cover this.

4.6.2 Word Filing

Word filing is a broad sense of the activities carried out here. Basically, each word or entity is reduced to a single figure or alphabet or even a symbol as seen in Figure 4.10. For convenience, we pick numbers. So, all the words noticed by the indexer are serially jotted down as a number. This way we can keep track of the word efficiently. It decreases the burden off the system.

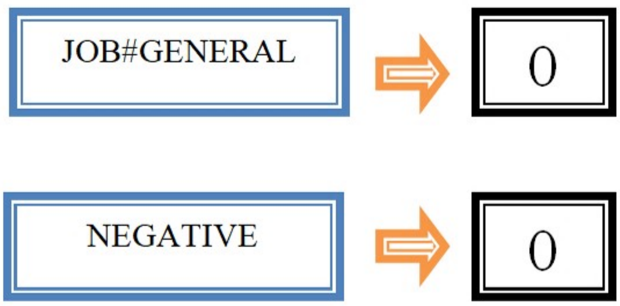
Figure 4.10: Word Filing



4.6.3 Class encryption

Class encryption is a scheme comparable to that of word filing but rather than acting on input data, it acts upon the output data as seen in Figure 4.11. This will lead to all the data being on the same page. There are binary and sequential encryptions. We opted for sequential encryption as we didn't see the demand for a binary encryption which is a higher level encryption.

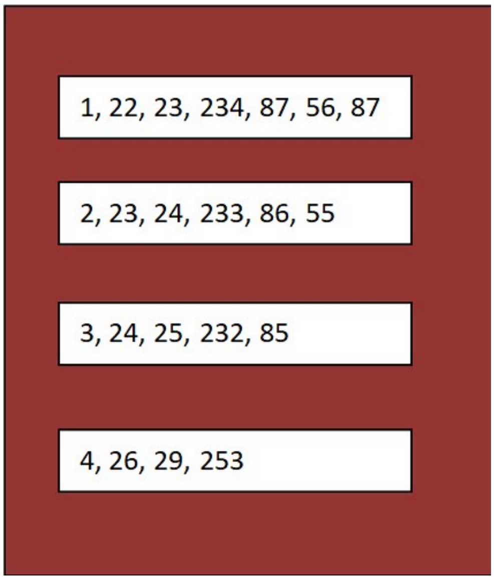
Figure 4.11: Class encryption



4.6.4 Arrangement

As explained earlier, all the words in the input data are indexed which means they are all bunch of integers as seen in Figure 4.12. We now had to replace all the words with numbers in the actual data. Subsequent to that, we have to put them into a data structure. They are lists of lists or in other words just sentences. This is left to the sequencing activity.

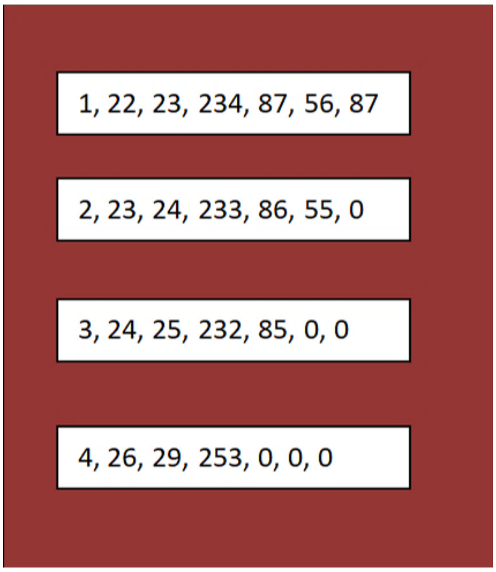
Figure 4.12: Arrangement



4.6.5 Filling

Filling step is just a formatting step. Though it does not influence the actual information, it influences the assembly of the information as seen in Figure 4.13. All the sentences ought to be of the same length. Naturally, that is not the case. Therefore we extend all the sentences to the measurement of the longest sentence by means of filler. In our situation, we elect zero as our filler.

Figure 4.13: Filling



4.7 Facet Establishment

Facet Establishment consists of only Token Embedding. It helps with pinpointing and gathering features that are present in the data.

4.7.1 Token Embedding

Token Embedding is a rising practice in today's implantations. It joins all the tokens that are densely linked to any of the already obtained tokens as seen in Figure 4.14. It results in the mounting of words and can be accessed by the system for reference. It produces a lexicon or a bank of words. It essentially augments the data than what would have been without token embedding.

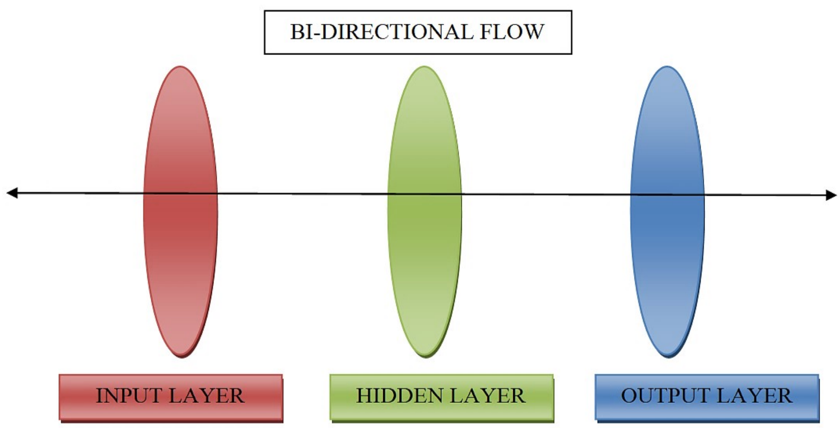
Figure 4.14: Token Embedding

4.8 Bi-Directional Gated Recurrent Unit

Generally, like any other deep learning framework, the direction of flow is from the start to the end. This indicates there is only one pathway. However, here the code comprises two pathways so that every word can have a relationship with every other word no matter how far apart they were.

Two hidden layers were opted sufficient as increasing the number of layers will not ameliorate the performance any further and decreasing the number of layers will have a negative impact on the same. The size is stated in the line of code along with switching on the return sequence functionality as in Figure 4.15.

Figure 4.15: Bi-Directional Gated Recurrent Unit



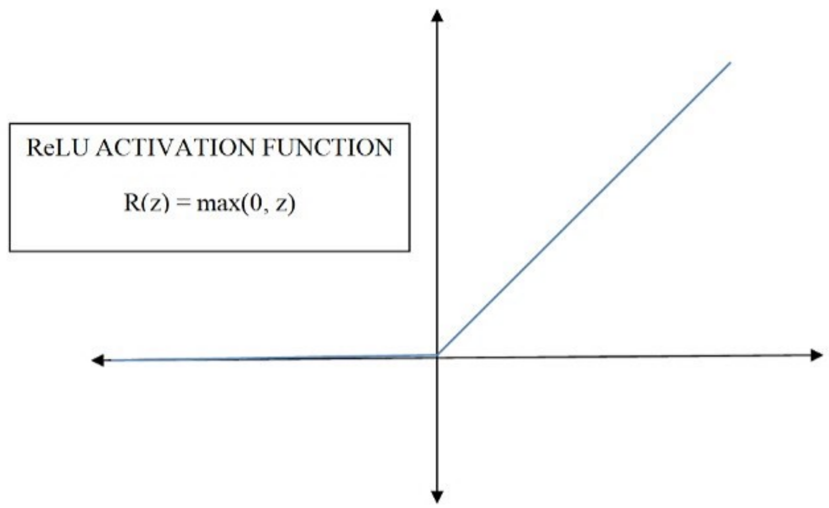
4.8.1 Activation Function

Activation function is crucial in binning the input into its respective output. Activation function has a threshold. Above or below which will fix on a certain result. It is a probabilistic function and hence it compares probabilities. There are many activation functions for various purposes. We have utilized two activation functions at distinctive layers.

4.4.1.1 Rectified Linear Units

Rectified Linear Units (ReLU) activation function is exercised at the hidden layers. This ranges from zero and a high range. It is a widely applied activation function. ReLU activation function was used by us in the dense layers as in Figure 4.16.

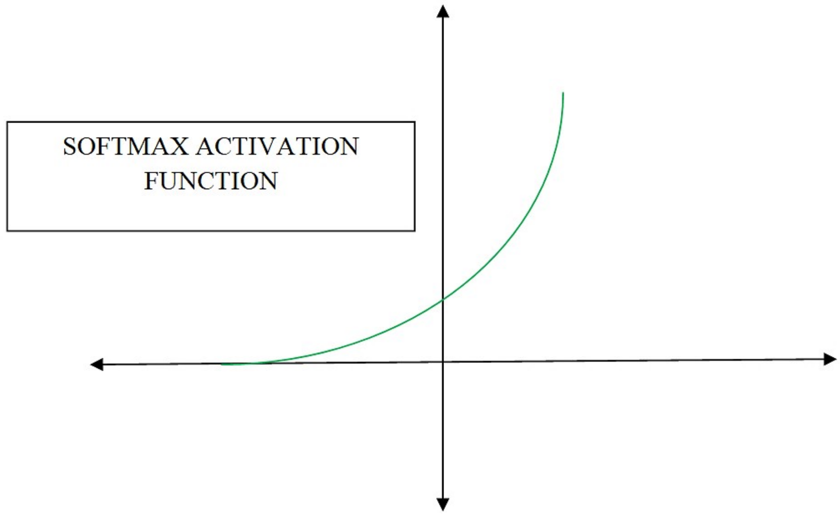
Figure 4.16: Rectified Linear Units



4.4.1.2 Softmax

Softmax activation function is exercised at the output layer. This ranges from zero and one. It is also a widely applied activation function at this precise layer. Softmax activation function is suitable for a classification problem as in Figure 4.17.

Figure 4.17: Softmax



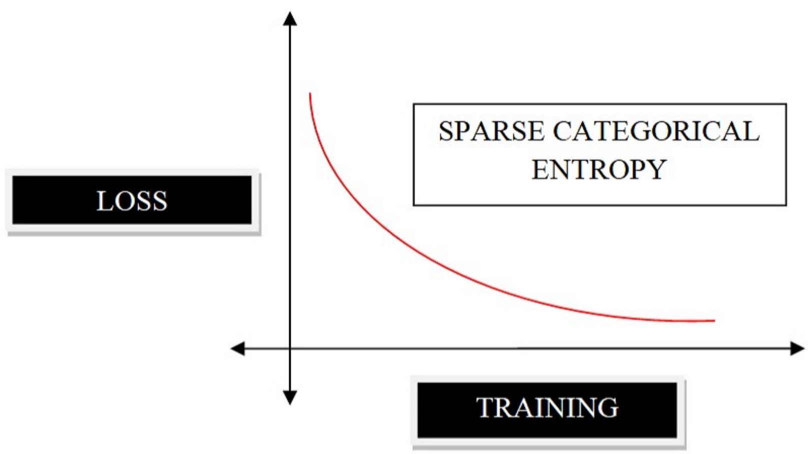
4.8.2 Loss Calculator

Loss function is the most vital measure in directing which way the model needs to head. The difference between the actual and the guessed approximations is known as the loss of the model. It will constantly work to minimize this discrepancy. Without this function, the model is walking through a dark night of blindness. Criticism and feedback about its work is mandatory.

4.4.2.1 Sparse Categorical Entropy

If there is sparse categorical entropy, there is also categorical entropy. There are several reasons why one would choose one over the other. The reason we chose this entropy is that our IDE recommended it to us due to autonomous classes as in Figure 4.18.

Figure 4.18: Sparse Categorical Entropy



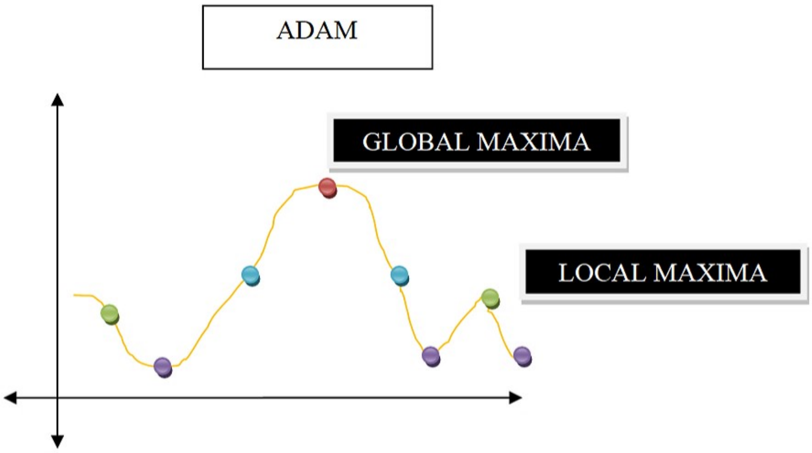
4.8.3 Optimizer

The optimizer selects the best path from all the probable paths. It is often not about cracking the solution. It is about doing it in the best way possible. We run behind efficiency. Optimization is the demand because there are many who can solve the case. People are looking to save their resources as much as physically possible before exhausting it.

4.4.3.1 Adam

Everyone is aware that the stochastic gradient descent is the premium selection for machine learning algorithms. However, for deep learning algorithms, Adam optimizer is highly advised as it believed to integrate the power of Stochastic Gradient Descent (SGD) and Root Mean Square Propagation (RMSprop) as in Figure 4.19.

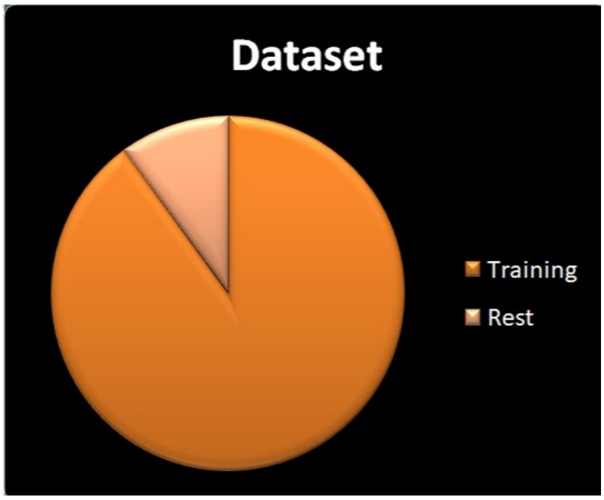
Figure 4.19: Adam



4.9 Training

In simple words, training is getting the paradigm used to what the user's intentions are which is clear from the dataset he wishes it to train as seen in Figure 4.8. In the dataset, depending on the number of output grades. It goes without saying that there must be enough instances for the paradigm to teach itself the pattern hidden behind the input which influences the output. We must also confirm that it is trained for long enough so that it is thorough and is ready to face any undetected case. After training, there will be testing. From the results of testing, we can then settle on whether to train more data or use a variant dataset or train extensively or stop learning midway when a threshold is reached. Hence, training is also a key step in the framework as in Figure 4.20.

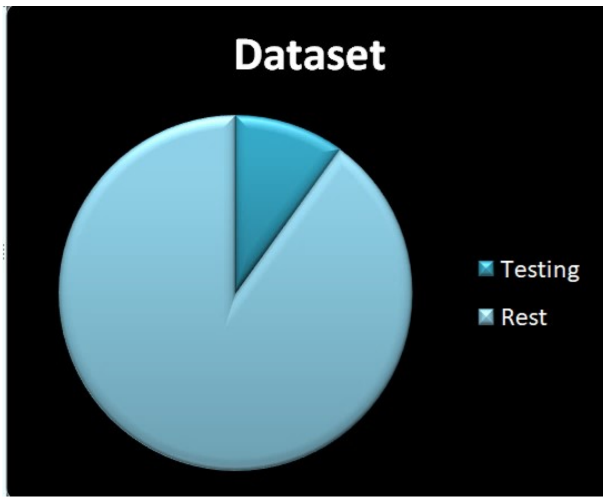
Figure 4.20: Training



4.10 Testing

Following training is the testing phase which gives the result of the whole program. Testing is equivalent to a report card. We scan the testing output to resolve any conflict that may be present. There are two impeding predicaments which are high bias and high variance. It can be paraphrased as overfitting and underfitting troubles.

Figure 4.21: Testing



To rectify overfitting, we add regularization to the cost function or add dropout to the model layers or train more data. To rectify underfitting, train for a longer time or add more parameters that increase complexity or boost the size of the model as seen in Figure 4.21.

CHAPTER 5

RESULT AND DISCUSSION

We tested our program and accomplished an accuracy of 92.5% for aspect establishment and 98% for sentiment classification. There are various metrics based on which the code was tested.

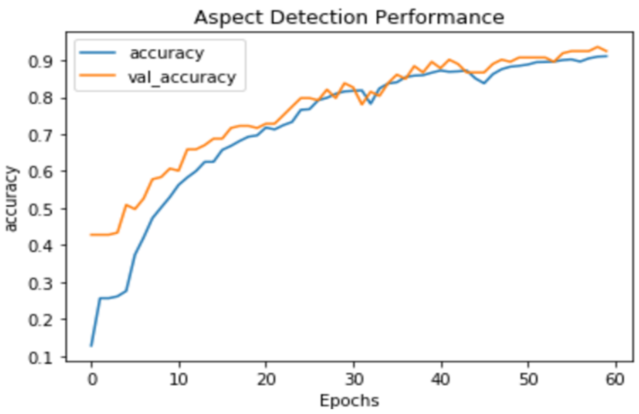
METRICS	VALUES (ASPECT ESTABLISHEMT/SENTIMENT CLASSIFICATION)
ACCURACY	92.5%/98%
TIME COMPLEXITY	4ms per sample (mean) ; 7s per batch (mean)
SPACE COMPLEXITY	134KB
PARAMETER SIZE	18,248/62,641
CONVERGENCE RATE	HIGH
OVERHEAD	LOW

Table 5.1: Testing

The time complexities of our framework are 4ms (mean) per sample and 7s (mean) per batch. The space complexity of our framework is 134KB. The parameter size is precisely 18,248 and 62,641 for aspect establishment and sentiment classification respectively. All of this is tabulated in Table 5.1.

Figure 5.1: Performance of Aspect Categorization

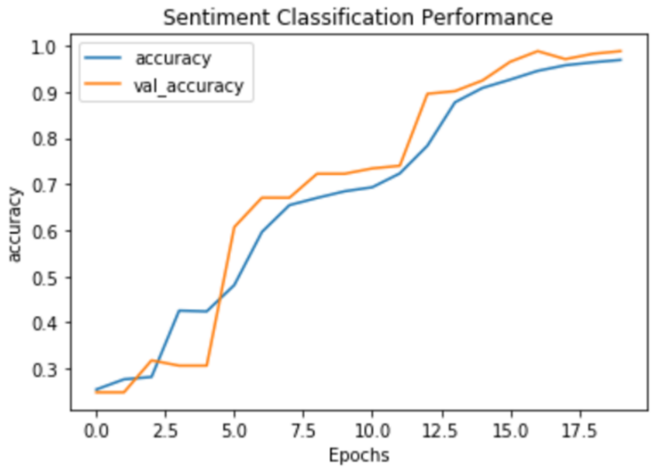
Epochs	Accuracy	Val_Accuracy
0	0.125	0.43
10	0.55	0.58
20	0.67	0.7
30	0.75	0.8
40	0.8	0.85
50	0.85	0.88
60	0.9	0.925



From the Figure in 5.1 and 5.2, the trends of training and validation accuracies concerning the number of epochs are represented with the help of a line graph for both Aspect Categorization and Sentiment Classification. All four curves are fairly smooth. This shows that there is not any instability in the training and testing phases. This may occur when the hyperparameters like learning are not set properly. It may be higher than required. The gap between the training and the testing curve is pretty low. This proves that there is not much bias and variance crisis. As indicated before, all the accuracies reach an appreciably elevated margin.

Figure 5.2: Performance for Sentiment Classification

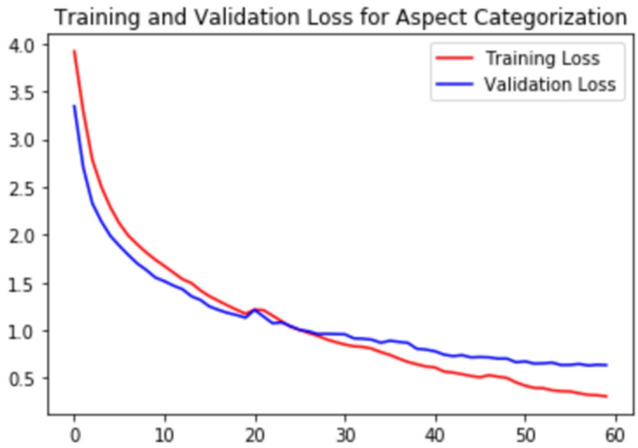
Epochs	Accuracy	Val_Accuracy
0.0	0.1	0.1
2.5	0.35	0.32
5.0	0.45	0.55
7.5	0.65	0.7
10.0	0.68	0.72
12.5	0.75	0.88
15.0	0.88	0.92
17.5	0.965	0.98



The output generated is the test data result. The test result data encompasses tags and probability distributions of Aspect Based Sentimental Analysis. The alternative service provided is customized user input. This means anyone can enter their review of a product and the result will be constructed for the same. We also supply a performance analysis of the algorithm as well as a comparative analysis with other challenging algorithms.

Figure 5.3: Cost of Aspect Categorization

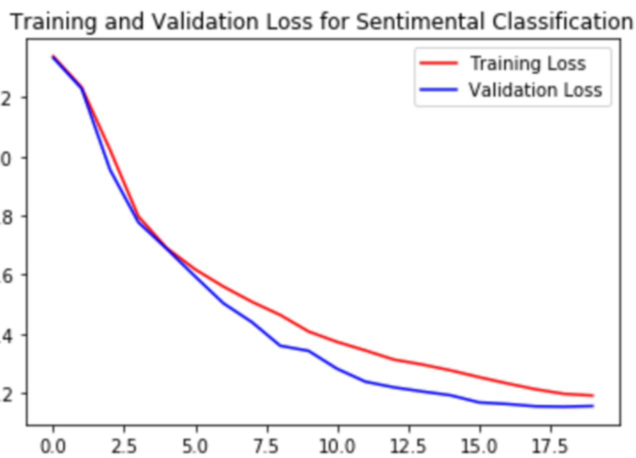
Epochs	Loss	Val_Loss
0	3.9	3.4
10	1.75	1.5
20	1.25	1.25
30	1.0	1.1
40	0.75	1.0
50	0.5	0.8
60	0.25	0.8



From Figure 5.3 and 5.4, the trends of training and validation loss concerning the number of epochs is represented with the help of a line graph for both Aspect Categorization and Sentiment Classification again. These curves are much smoother than the accuracy curves. This is a great sign as loss must decrease evenly. The loss diminishes less than 10% in the first case and less than 5% in the second case. This is quite commendable. There is not much space between the Training and Testing curves in both graphs. This illustrates that there is no prejudice or discrepancy in the model. The parameters and the hyperparameters are tuned consequently.

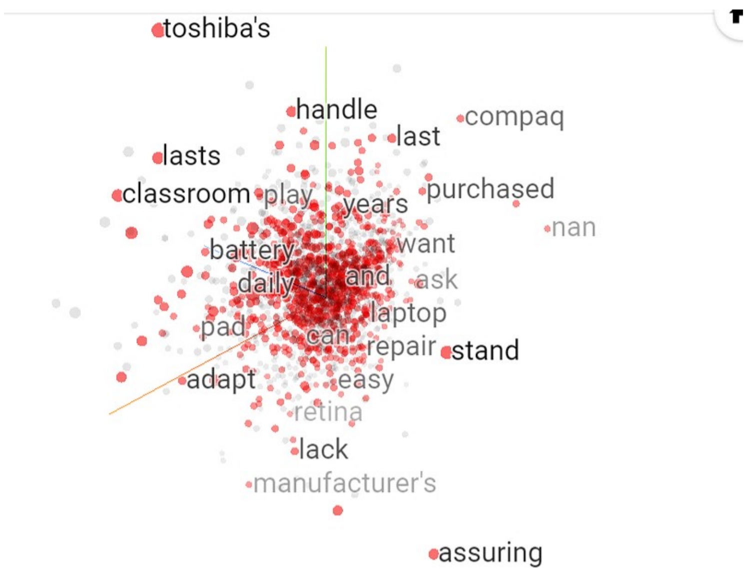
Figure 5.4: Cost of Sentimental Analysis

Epochs	Loss	Val_Loss
0.0	1.4	1.4
2.5	0.8	0.8
5.0	0.62	0.6
7.5	0.43	0.38
10.0	0.35	0.2
12.5	0.3	0.18
15.0	0.28	0.15
17.5	0.25	0.12



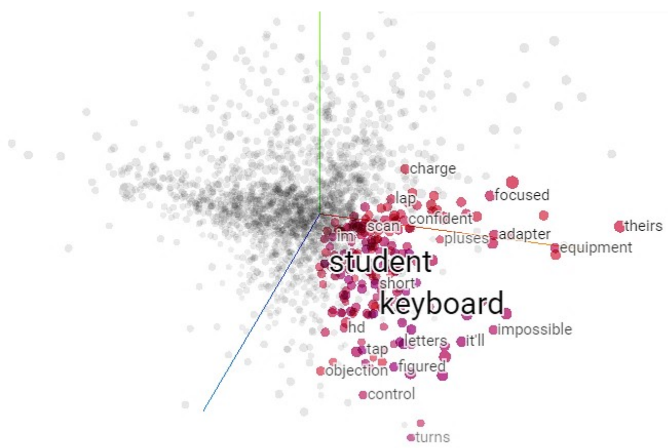
From Figure 5.5, the embedding projection was worked out and the Principle Component Analysis is projected. This is the embedding visualization of Aspect Categorization. All the words and their neighbors are depicted.

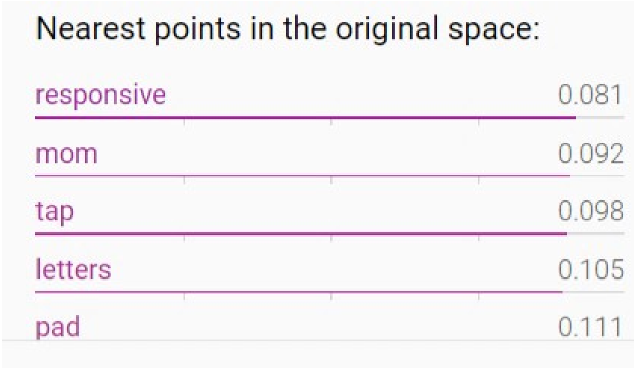
Figure 5.5: Embedding Projection for Aspect Categorization



From Figure 5.6, a specific aspect class called 'Keyboard' was picked. All its closely related words and their cosine distance are displayed. Few words are in a shorter distance than the others. The word 'Pad' and 'Letters' are closer to 'Keyboard'. Even the words 'Student', 'Charge', 'Control', and 'Equipment' are in closer proximity.

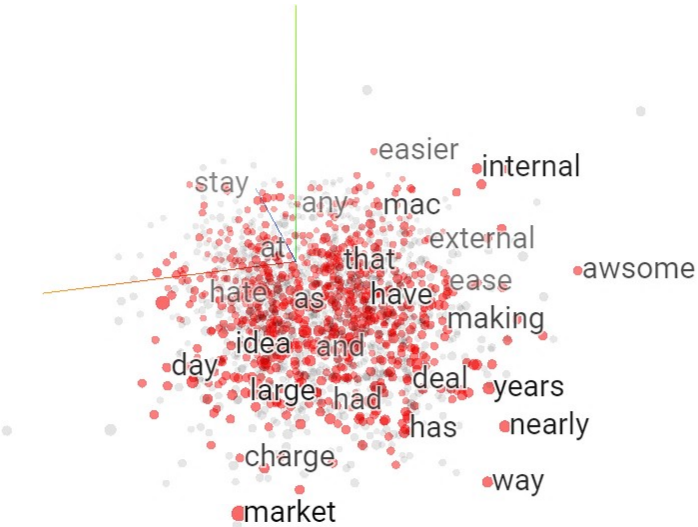
Figure 5.6: Embedding Project for Keyboard Aspect





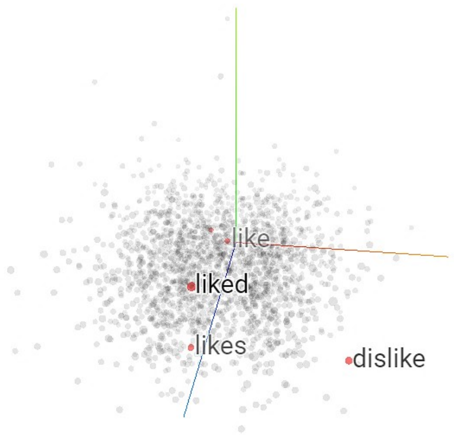
Similarly from Figure 5.7, the embedding projection was worked out and the Principle Component Analysis is projected for Sentiment Classification embedding visualization. All the words and their neighbors are depicted again.

Figure 5.7: Embedding Projection for Aspect Categorization



From Figure 5.8, a specific sentiment called 'Like' was picked. The words 'Likes' and 'Liked' are nearby whereas the word 'Dislike' is very far away compared to the other two which shows by sentiment class, all the negative, positive, and neutral words are formed into organizations like clusters.

Figure 5.8: Embedding Project for Like Sentiment



Data Visualization gives an imminent vision that was not there before. It is tough to scrutinize that massive quantity of information and put people in the dark about potential and inherent tribulations. So, Data Visualization posses as an elucidation to this quandary.

The experiment incorrectly classified an example whose aspect category was 'Design' to 'Software'. The lack of enough data might have not given the framework much evidence to take a misinformed stand. All these are expected to be rectified during revisions for the next version.

CHAPTER 6

CONCLUSION

From this work, it is concluded that a novel architecture for Aspect based Sentimental Analysis was demonstrated. The accuracies of famous Machine Learning mechanisms and state of the art Deep Learning methods were challenged. Not only Sentimental Analysis but Aspect Based Sentimental Analysis was also incorporated properly. The code was weighed against other techniques like RNN, Support Vector Machine, Deep Neural Networks, and LSTM. GRU proved to cross the values of all other algorithms in many gauging metric areas. The program improved its prediction on test data when it's learning on the training data kept increasing and became thorough. But once it has pretty much figured out the hidden maps in the input data, it did not show a hike in test accuracy anymore as it is not encountering new samples. Many people include positive and negative sentiments but they forget that certain sentences or neutral or do not have any sentiment to it. Even those types of text were introduced to it so that it is familiar with that kind of phrase as well. Though a thorough fine structure was not put together, its capabilities and utilities are expected to be expanded.

There are so many ideas that come to mind for enhancing the existing model in the future. Firstly, trying to collect a huge amount of web scrapped data when there is enough time to produce a new dataset. Secondly, a permutation of algorithms in each module can be juxtaposed and the differences that are made to the structure can be examined. Thirdly, append an aspect clustering phase so that it becomes effortless while navigation. Fourthly, combining other sub-domains of ABSA like summarization and cross-domain examination is possible which will make it all-purpose and all-round. Finally, there is a strong intention to make it inclusive realms and sellers of the same item to that is more so demonstrative for customers so that their work is decreased to a great extent. If one seeks out ways to cross the boundaries set by others and explore new areas. There are countless concepts out there that are not harnessed yet. This sets apart one from the rest. It takes a lot more than knowledge to do that, it takes patience.

REFERENCES

[1] Augustyniak, u., Kajdanowicz, T., and Kazienko, P. "Aspect detection using word and char embeddings with BI-LSTM and crf. 43–50, 2019.

[2] Chen, J., Hou, H., Ji, Y., and Gao, J. "Graph convolutional networks with structural attention model for aspect-based sentiment analysis." *2019 International Joint Conference on Neural Networks (IJCNN)*. 1–7, 2019.

[3] Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. "On the properties of neural machine translation: Encoder-decoder approaches." 2014.

[4] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. "Empirical evaluation of gated recurrent neural networks on sequence modeling." 2014.

[5] Jiang, M., Wu, J., Shi, X., and Zhang, M. "Transformer based memory network for sentiment analysis of web comments." *IEEE Access,* 7, 179942–179953, 2019.

[6] Khine, W. L. K. and Aung, N. T. T. "Applying deep learning approach to targeted aspect-based sentiment analysis for restaurant domain." *2019 International Conference on Advanced Information Technologies (ICAIT)*. 206–211, 2019.

[7] Kumar, A., Narapareddy, V. T., Aditya Srikanth, V., Neti, L. B. M., and Malapati A. "Aspect-based sentiment classification using interactive gated convolutional network." *IEEE Access,* 8, 22445–22453, 2020.

[8]. Luo, H., Li, T., Liu, B., Wang, B., and Unger, H. "Improving aspect term extraction with bidirectional dependency tree representation." 2018.

[9] Lv, Y., Hu, M., Yang, C., Tang, Y., and Wang, H. "Extract, attend, predict: Aspect-based sentiment analysis with deep self-attention network." *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS).* 297–304, 2019.

[10] Mary, A. J. J. and Arockiam, L. "Asful: Aspect-based sentiment summarization using fuzzy logic." *2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET)*, 1–5, 2017.

[11] Meng, W., Wei, Y., Liu, P., Zhu, Z., and Yin, H. "Aspect-based sentiment analysis with feature enhanced attention cnn-bilstm." *IEEE Access, 7,* 167240–167249, 2019.

[12] Onaciu, A. and Marginean, A. "Ensemble of artificial neural networks for aspect-based sentiment analysis." 13–19, 2018.

[13] Rana, T. A. and Yu-N Cheah "Hybrid rule-based approach for aspect extraction and categorization from customer reviews." *2015 9th International Conference on IT in Asia (CITA)*. 1–5, 2015.

[14] Schouten, K., van derWeijde, O., Frasincar, F., and Dekker, R. "Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data." *IEEE Transactions on Cybernetics*, 48(4), 1263–1275, 2018.

[15] Son, L. H., Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., and Abdel-Basset, M. "Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network." *IEEE Access*, 7, 23319–23328, 2019.

[16] Wang, Y., Chen, Q., Ahmed, M., Li, Z., Pan, W., and Liu, H. "Joint inference for aspect-level sentiment analysis by deep neural networks and linguistic hints." *IEEE Transactions on Knowledge and Data Engineering,* 1–1, 2019.

[17] Xiang, C., Ren, Y., and Ji, D. "Identifying implicit polarity of events by using an attention-based neural network model." *IEEE Access*, 7, 133170–133177, 2019.

[18] Xie, J., Chen, B., Gu, X., Liang, F., and Xu, X. "Self-attention-based BILSTM model for short text fine-grained sentiment classification." *IEEE Access*, 7, 180558–180570, 2019.

[19] Yan, D., Chen, J., Cui, J., Shan, A., and Shi, W. "Deep multi-head attention network for aspect-based sentiment analysis." *2019 IEEE International Conference on Big Data (Big Data)*. 695–700, 2019.

[20] Yeo, I. and Balachandran, K. "Sentiment analysis on time-series data using weight priority method on deep learning." *2019 International Conference on Data Science and Communication (IconDSC)*. 1–7, 2019.

[21] Zeng, J., Ma, X., and Zhou, K. "Enhancing attention-based LSTM with position context for aspect-level sentiment classification." *IEEE Access*, 7, 20462–20471, 2019.

[22] Zhou, Z., Liu, F., and Wang, Q. "R-transformer network based on position and self-attention mechanism for aspect-level sentiment classification." *IEEE Access*, 7, 127754–127764, 2019.

[23] Kedar Potdar, Taher S. Pardawala, Chinmay D. Pai, "A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers." *In proc. of International Journal of Computer Applications (0975 – 8887)*, Volume 175 – No.4, 2017.

APPENDIX A

#Import Packages

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import keras
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

#Read Dataset

```
data = pd.read_csv('Train SemEval 2015 Task 12.csv')
test = pd.read_csv('Test SemEval 2015 Task 12.csv')
data1 = pd.read_csv('Train extended.csv')
data1
```

#Reviews and Aspect Labels

```
train_sen1 = data.iloc[:,0]
train_sen2 = data.iloc[:,3]
train_sen3 = data.iloc[:,6]
train_sen4 = data1.iloc[:,2]
train_sen5 = data1.iloc[:,6]
test_sen = test.iloc[:,10]
test_aspect = test.iloc[:,11]
aspect_sen1 = data.iloc[:,1]
aspect_sen2 = data.iloc[:,4]
aspect_sen3 = data.iloc[:,7]
```

```
aspect_sen4 = data1.iloc[:,3]
aspect_sen5 = data1.iloc[:,7]
aspect_sen = aspect_sen1.append(aspect_sen2)
aspect_sen = aspect_sen.append(aspect_sen3)
aspect_sen = aspect_sen.append(aspect_sen4)
aspect_sen = aspect_sen.append(aspect_sen5)
aspects = aspect_sen.append(test_aspect)
train_sen = train_sen1.append(train_sen2)
train_sen = train_sen.append(train_sen3)
train_sen = train_sen.append(train_sen4)
train_sen = train_sen.append(train_sen5)
train_sen = train_sen.astype(str)
test_sen = test_sen.astype(str)
aspects = aspects.astype(str)
aspects = aspects.astype('category')
aspects = aspects.cat.codes
aspects_train = aspects[:1731]
aspects_test = aspects[1731:]
aspects_train = np.asarray(aspects_train)
aspects_test = np.asarray(aspects_test)


    #Tokenization
tokenizer = Tokenizer(num_words=2100, oov_token='<OOV>')
tokenizer.fit_on_texts(train_sen)
w_i = tokenizer.word_index
print(w_i)


    #Sequencing
seq_train = tokenizer.texts_to_sequences(train_sen)
seq_test = tokenizer.texts_to_sequences(test_sen)
print(seq_test)
```

#Padding

```
pad_train = pad_sequences(seq_train, padding='post',maxlen=100)
pad_test = pad_sequences(seq_test, padding='post',maxlen=100)
print(pad_test)
```

#Sentiment Labels

```
lab_train1 = data.iloc[:,2]
lab_train2 = data.iloc[:,5]
lab_train3 = data.iloc[:,8]
lab_train4 = data1.iloc[:,4]
lab_train5 = data1.iloc[:,8]
lab_train = lab_train1.append(lab_train2)
lab_train = lab_train.append(lab_train3)
lab_train = lab_train.append(lab_train4)
lab_train = lab_train.append(lab_train5)
lab_test = test.iloc[:,12]
labs = lab_train.append(lab_test)
lab_train = lab_train.astype(str)
lab_test = lab_test.astype(str)
labs = labs.astype(str)
labs = labs.astype('category')
labs = labs.cat.codes
labs_train = labs[:1731]
labs_test = labs[1731:]
labs_train = np.asarray(labs_train)
labs_test = np.asarray(labs_test)
labs
```

#Model1 for Aspect Categorization - GRU

```
model1 = tf.keras.Sequential([ tf.keras.layers.Embedding(2100,24, input_length=100),
```

```
tf.keras.layers.Bidirectional(tf.keras.layers.GRU(24,return_sequences=True)),
tf.keras.layers.Bidirectional(tf.keras.layers.GRU(12)),
tf.keras.layers.Dense(6,activation='relu'),
tf.keras.layers.Dense(61,activation='softmax') ])
model1.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model1.summary()
```

#Model2 for Sentiment Classification - GRU

```
model2 = tf.keras.Sequential([ tf.keras.layers.Embedding(2100,8, input_length=100),
tf.keras.layers.Bidirectional(tf.keras.layers.GRU(8,return_sequences=True)),
tf.keras.layers.Bidirectional(tf.keras.layers.GRU(4)),
tf.keras.layers.Dense(4,activation='relu'),
tf.keras.layers.Dense(4,activation='softmax') ])
model2.compile(loss='sparse_categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
model2.summary()
```

#Training and Prediction of Model1

```
epoch1 = 60
history1 = model1.fit(pad_train, aspects_train, epochs=epoch1,
validation_data=(pad_test,aspects_test))
pred1 = model1.predict(pad_test)
print(pred1)
print(aspects_test)
```

#Training and Prediction for Model2

```
epoch2 = 20
history2 = model2.fit(pad_train, labs_train, epochs=epoch2, validation_data=(pad_test,labs_test))
pred2 = model2.predict(pad_test)
print(pred2)
print(lab_test)
```

#Performance Visualization - Accuracy for Aspect Categorization

```
def plot_graphs1(history1, string):
plt.plot(history1.history[string])
plt.plot(history1.history['val_'+string])
plt.xlabel("Epochs")
plt.ylabel(string)
plt.legend([string, 'val_'+string])
plt.title('Aspect Detection Performance')
plt.show()
plot_graphs1(history1, 'accuracy')
```

#Performance Visualization - Accuracy for Sentiment Classification

```
def plot_graphs2(history2, string):
plt.plot(history2.history[string])
plt.plot(history2.history['val_'+string])
plt.xlabel("Epochs")
plt.ylabel(string)
plt.legend([string, 'val_'+string])
plt.title('Sentiment Classification Performance')
plt.show()
plot_graphs2(history2, 'accuracy')
```

#Test on single random instance

```
instance = "I hate the software of the laptop"
instance = tokenizer.texts_to_sequences(instance)
flat_list = []
for sublist in instance:
for item in sublist:
flat_list.append(item)
flat_list = [flat_list]
instance = pad_sequences(flat_list, padding='post', maxlen=100)
```

#Aspect

```
model1.predict(instance)
```

#Sentiment

```
model2.predict(instance)
```

#For Comparison

```
print(test_aspect)
print(labs_test)
```

APPENDIX B

Figure 9.1: Dataset Output

	sentences/sentence/0/text	sentences/sentence/0/_id	sentences	aspects	polarity	sentences/sentence/1/_id
0	Being a PC user my whole life....	79:00:00	This computer is absolutely AMAZING!!!	LAPTOP#GENERAL	positive	79:01:00
1	the laptop was really good and it goes really ...	10:00	i would really recommend to any person out the...	LAPTOP#GENERAL	positive	10:01
2	As a lifelong Windows user, I was extremely pl...	264:00:00	As a computer science student in college, I fi...	LAPTOP#PORTABILITY	positive	264:01:00
3	Oh my goodness-I am not a happy camper.	24:00:00	My HP is very heavy.	LAPTOP#DESIGN_FEATURES	negative	24:01:00
4	Since I purchased my Toshiba netbook, I have b...	277:00:00	The netbook is easier for me to take to bed an...	LAPTOP#PORTABILITY	positive	277:01:00
...	...	...	...	...	...	...
445	This laptop is amazing!	B00KB3MXH4_22_A106YGESUYA4BP:0	Windows 8.1 has its	OS#GENERAL	neutral	B00KB3MXH4_22_A106YGESUYA4BP:1

Figure 9.2: Tokenization and Word Filing Output

{'<OOV>': 1, 'the': 2, 'i': 3, 'it': 4, 'and': 5, 'to': 6, 'a': 7, 'is': 8, 'for': 9, 'of': 10, 'this': 11, 'my': 12, 'have':
13, 'laptop': 14, 'that': 15, 'with': 16, 'computer': 17, 'was': 18, 'not': 19, 'in': 20, 'nan': 21, 'on': 22, 'very': 23, 'b
ut': 24, 'use': 25, 'great': 26, 'had': 27, 'so': 28, 'you': 29, 'has': 30, 'as': 31, 'one': 32, 'all': 33, 'easy': 34, 'me':
35, 'only': 36, 'battery': 37, 'good': 38, 'just': 39, 'bought': 40, 'up': 41, 'be': 42, 'after': 43, 'do': 44, 'are': 45, 'm
acbook': 46, 'love': 47, 'its': 48, 'well': 49, 'no': 50, 'screen': 51, 'more': 52, 'am': 53, "it's": 54, 'life': 55, 'toshib
a': 56, 'like': 57, 'about': 58, 'mac': 59, 'from': 60, 'get': 61, 'than': 62, 'would': 63, 'time': 64, 'really': 65, 'window
s': 66, 'first': 67, 'pro': 68, 'or': 69, 'out': 70, 'they': 71, 'everything': 72, 'product': 73, 'price': 74, 'can': 75, 'a
t': 76, 'any': 77, 'pc': 78, 'work': 79, 'will': 80, '2': 81, 'an': 82, 'quality': 83, 'when': 84, 'months': 85, 'keyboard':
86, "i've": 87, 'if': 88, 'what': 89, 'then': 90, 'we': 91, 'fast': 92, 'hard': 93, 'drive': 94, 'got': 95, 'because': 96, 'p
roblems': 97, "don't": 98, 'buy': 99, 'never': 100, 'ever': 101, 'machine': 102, 'speed': 103, 'much': 104, 'need': 105, 'lon
g': 106, 'back': 107, 'since': 108, 'purchased': 109, '1': 110, "i'm": 111, 'laptops': 112, 'apple': 113, 'been': 114, 'littl
e': 115, 'used': 116, 'years': 117, 'even': 118, '7': 119, 'also': 120, 'thing': 121, 'best': 122, 'does': 123, 'which': 124,
'by': 125, 'perfect': 126, 'new': 127, 'works': 128, 'needs': 129, 'still': 130, 'system': 131, 'power': 132, 'software': 13
3, 'there': 134, 'slow': 135, 'netbook': 136, 'graphics': 137, 'dell': 138, 'some': 139, '3': 140, 'over': 141, 'how': 142,
'other': 143, 'features': 144, 'were': 145, 'went': 146, 'hp': 147, 'awesome': 148, 'now': 149, "can't": 150, 'pad': 151, 'no
tebook': 152, 'could': 153, 'size': 154, 'another': 155, 'working': 156, 'better': 157, 'way': 158, 'happy': 159, 'amazing':
160, 'bad': 161, 'right': 162, 'day': 163, 'money': 164, 'display': 165, 'always': 166, 'most': 167, 'problem': 168, 'excelle
nt': 169, 'owned': 170, 'want': 171, 'nice': 172, 'did': 173, 'needed': 174, 'too': 175, 'performance': 176, 'home': 177, 'to
p': 178, 'around': 179, 'internet': 180, 'make': 181, 'their': 182, '5': 183, 'able': 184, 'light': 185, 'enough': 186, 'game

Figure 9.3: Arrangement Output

[[1963, 41, 476, 945, 9, 1964], [24, 88, 29, 99, 11, 3, 63, 1535, 258, 15, 29, 99, 7, 400, 5, 51, 1521, 9, 11, 102, 31, 71, 45,
1965, 6, 42, 317, 1966], [21], [2, 357, 1536, 226, 478, 4, 7, 409, 590, 62, 467, 46, 68, 575, 15, 116, 7, 1967, 93, 94], [21],
[3, 75, 25, 4, 388], [1968, 454, 22, 2, 421, 718, 752, 124, 111, 130, 302, 6, 61, 6, 2, 1969, 10], [128, 23, 49], [92, 186, 9,
12, 911], [2, 273, 224, 16, 11, 18, 66, 119], [21], [2, 539, 165, 8, 639], [8, 39, 160], [21], [1537, 57, 357, 1536], [21], [2
1], [21], [3, 512, 118, 671, 1970, 22, 4], [148], [54, 19, 89, 29, 802, 9], [21], [21], [21], [1971, 1972, 138, 1973], [21], [2
1], [2, 36, 836, 3, 300, 485, 4, 7, 183, 1538, 8, 3, 241, 3, 13, 6, 920, 28, 194, 6, 456, 22, 2, 210, 447], [38, 73, 9, 7, 38,
74], [21], [3, 344, 101, 99, 7, 17, 411, 933, 205, 931, 9, 584], [71, 145, 169, 6, 1528, 221, 434, 5, 870, 221, 1974, 107, 229,
221, 1975, 2, 349, 163], [71, 546, 4], [63, 19, 258], [98, 487, 6, 25, 6, 289, 77, 78, 187, 69, 455], [12, 1976, 18, 6, 19, 56
2, 11, 32, 50, 295, 5, 50, 599, 6, 1511, 12, 1977, 514, 6, 1532, 724], [39, 536], [88, 29, 45, 7, 843, 69, 105, 4, 9, 386, 17,
25, 61, 257, 537], [21], [2, 86, 8, 7, 115, 1978, 16, 218, 6, 25, 2, 1510, 781, 6, 61, 2, 1979, 433, 24, 16, 142, 1980, 3, 25,
866, 4, 8, 7, 927, 359], [1981, 92, 186], [75, 118, 1982, 1983, 22, 11, 17], [21], [452, 11, 17, 616, 16, 7, 629, 154, 86], [2
1], [21], [21], [1984, 484], [126, 154, 9, 35], [21], [54, 185, 5, 34, 6, 25, 249], [21], [21], [120, 23, 473, 14], [2, 37, 55,
8, 38, 118, 16, 223, 297, 1985, 133, 57, 1986, 119, 9, 7, 38, 1987, 10, 2, 163], [21], [21], [76, 12, 1988, 3, 105, 11, 366, 1
0, 14], [21], [21], [538, 595, 246, 342], [640, 7, 83, 14], [21], [21], [21], [19, 7, 1989, 24, 257, 87, 331, 6, 532], [2, 539,
51, 8, 544, 756, 5, 2, 306, 8, 2, 122, 306, 6, 751], [154, 8, 126, 5, 165, 8, 756], [2, 51, 8, 1513, 5, 845, 169, 663], [11, 1
4, 123, 19, 13, 82, 1990, 1991, 925, 752, 9, 1992, 496, 908], [21], [12, 943, 921, 11, 14, 283, 5, 451, 42, 58, 4, 5, 3, 325,
4, 18, 7, 38, 224, 5, 40, 4], [134, 8, 7, 1993, 329, 8, 378, 24, 907, 111, 19, 206, 6, 487, 155, 32, 11, 8, 7, 901, 73], [21],
[21], [3, 80, 42, 1520, 4], [2, 240, 8, 884], [4, 123, 13, 7, 1994, 10, 1995, 445, 84, 29, 487, 6, 44, 2, 1996, 20, 1997, 24, 1
41, 33, 644, 23, 380], [3, 13, 120, 116, 4, 6, 601, 1998, 5, 1999], [21], [21], [3, 53, 223, 2000, 9, 322, 5, 2001, 204, 111, 2
2, 2, 1522], [21], [3, 300, 13, 15, 104, 164, 28, 3, 325, 1530, 485, 2, 2002, 7, 487], [3, 44, 7, 409, 10, 238, 210, 28, 3, 15
0, 42, 2003, 179, 76, 2, 178, 193, 9, 2004], [3, 244, 2005, 94, 16, 2006, 585, 2007, 94, 5, 8, 130, 317, 135], [3, 40, 4, 9, 1
2, 526, 1539, 2008, 5, 71, 47, 4, 2009, 2, 86, 5, 51], [18, 34, 6, 61, 41, 5, 611, 5, 616, 16, 66, 119, 68, 12, 674, 235, 131,
28, 194, 44, 19, 57, 66, 239], [3, 13, 2010, 2011, 9, 4, 39, 20, 400, 4, 890], [3, 63, 1535, 258, 11, 17, 4, 8, 92], [5, 323, 1

Figure 9.4: Filling Output

```
[[1963   41  476 ...    0    0    0]
 [  24   88   29 ...    0    0    0]
 [  21    0    0 ...    0    0    0]
 ...
 [  66  239    8 ...    0    0    0]
 [  74    8   26 ...    0    0    0]
 [  21    0    0 ...    0    0    0]]
```

Figure 9.5: Class Encryption Output

```
0      0
1      3
2      0
3      1
4      3
      ..
168    3
169    0
170    1
171    3
172    0
Length: 1904, dtype: int8
```

Figure 9.6: Model for Aspect Establishment Output

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 100, 24)           50400
_________________________________________________________________
bidirectional (Bidirectional (None, 100, 48)           7200
_________________________________________________________________
bidirectional_1 (Bidirection (None, 24)                4464
_________________________________________________________________
dense (Dense)                (None, 6)                 150
_________________________________________________________________
dense_1 (Dense)              (None, 61)                427
=================================================================
Total params: 62,641
Trainable params: 62,641
Non-trainable params: 0
_________________________________________________________________
```

Figure 9.7: Model for Sentiment Classification Output

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 100, 8)            16800
_________________________________________________________________
bidirectional_2 (Bidirection (None, 100, 16)           864
_________________________________________________________________
bidirectional_3 (Bidirection (None, 8)                 528
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 36
_________________________________________________________________
dense_3 (Dense)              (None, 4)                 20
=================================================================
Total params: 18,248
Trainable params: 18,248
Non-trainable params: 0
_________________________________________________________________
```

Figure 9.8: Accuracy and Output for Model 1

```
1731/1731 [==============================] - 7s 4ms/sample - loss: 0.3433 - accuracy: 0.9110 - val_loss: 0.5624 - val_accurac
y: 0.9249
[[2.4608416e-11 3.2769414e-08 3.4175455e-14 ... 3.1679849e-07
  3.0076404e-16 9.8548293e-01]
 [4.7966719e-09 2.5197564e-06 3.9358938e-09 ... 1.0595185e-03
  3.2049114e-12 2.9073947e-03]
 [1.1676092e-11 4.0911474e-09 2.6017636e-15 ... 2.9257484e-08
  9.0665004e-17 9.9583673e-01]
 ...
 [6.8414870e-08 6.3937233e-07 5.4171010e-14 ... 4.6361195e-13
  3.6534089e-05 2.2629829e-10]
 [2.7155805e-10 4.0883553e-04 4.2185529e-11 ... 7.4437817e-10
  2.7111560e-06 2.2506694e-13]
 [1.4580198e-11 4.5979354e-09 1.8490708e-15 ... 2.1563958e-08
  1.4943353e-16 9.9559432e-01]]
[25 28 60 60 60 26 21 25 25 43 60  8 23 60 28 60 60 60 29 23 23 60 60 60
 24 60 60 23 23 60 60 58 23 23 24 22 21 24 60 17 60 24 60 17 60 60 60  4
 22 60 22 60 60 25  2 60 60 24 60 60 21 28 60 60 60 22  8 22  7 22 60 60
 23 60 60 23 40 25 60 60 60 60 60 60 60 25 23 60 60 23  6 60 60 60 10 23
```

Figure 9.9: Accuracy and Output for Model 2 Part 1

```
Epoch 20/20
1731/1731 [==============================] - 7s 4ms/sample - loss: 0.2225 - accuracy: 0.9694 - val_loss: 0.1557 - val_accurac
y: 0.9884
[[1.27753607e-04 8.63221526e-01 8.45586783e-07 1.36649951e-01]
 [2.32009130e-04 8.15347314e-01 1.52376731e-06 1.84419155e-01]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
 [8.78906965e-01 2.52431259e-04 2.07022838e-02 1.00138403e-01]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
 [3.16578858e-02 7.12193698e-02 1.88601916e-04 8.96934152e-01]
 [1.28810119e-04 8.62983584e-01 8.56698080e-07 1.36886746e-01]
 [1.01715170e-01 1.68234501e-02 5.87332412e-04 8.80873978e-01]
 [3.06793954e-02 7.37191811e-02 1.82990683e-04 8.95418465e-01]
 [4.69885692e-02 4.54639606e-02 2.77284620e-04 9.07270253e-01]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
 [3.36159244e-02 6.65634349e-02 1.99614558e-04 8.99621010e-01]
 [4.06441718e-01 3.57050402e-03 3.51311918e-03 5.86474717e-01]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
 [2.42602434e-02 9.44453478e-02 1.45653757e-04 8.81148696e-01]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
 [9.20112133e-01 1.04413499e-04 3.04671507e-02 4.93162759e-02]
```

Figure 9.10: Accuracy and Output for Model 2 Part 2

```
 [8.59838605e-01 2.83495843e-04 1.67723633e-02 1.23105504e-01]
 [7.18361407e-04 6.83268487e-01 4.62901562e-06 3.16008538e-01]
 [9.01437342e-01 1.64456709e-04 6.66241273e-02 3.17740925e-02]
 [5.81737980e-02 3.52383405e-02 3.42648971e-04 9.06245232e-01]
 [9.06143606e-01 1.38232324e-04 2.30307169e-02 7.06875026e-02]
 [1.23152291e-04 8.64276290e-01 7.97823645e-07 1.35599717e-01]
 [3.48928608e-02 6.41583800e-02 2.07941033e-04 9.00740802e-01]
 [9.20112133e-01 1.04413703e-04 3.04671451e-02 4.93163317e-02]]
0      negative
1      negative
2           nan
3           nan
4           nan
         ...
168    positive
169         nan
170    negative
171    positive
172         nan
Name: sentences/sentence/2/Opinions/Opinion/0/_polarity, Length: 173, dtype: object
```

Figure 9.11: Output for User Instance Part 1

```
array([[8.2788920e-10, 2.7896150e-04, 9.1375291e-13, 3.6239651e-08,
        1.2992045e-01, 3.9591461e-05, 1.1109589e-04, 5.0467910e-09,
        1.1476715e-08, 1.7602831e-09, 8.8363699e-09, 1.5534883e-06,
        1.5233607e-03, 2.0215374e-17, 8.1224937e-07, 1.4497968e-08,
        8.8749864e-08, 5.4294447e-12, 7.1123033e-04, 3.2583015e-05,
        2.3049964e-03, 2.8745734e-04, 6.6866941e-04, 1.0798894e-11,
        3.8865522e-01, 4.2606704e-04, 4.7236600e-07, 2.5189885e-01,
        4.6823369e-04, 1.6230765e-01, 2.4915560e-06, 3.0160103e-02,
        1.3845727e-06, 4.6022187e-08, 2.3563633e-04, 3.0604122e-07,
        2.1625337e-05, 8.9505361e-03, 1.2134950e-03, 3.4977682e-05,
        2.0437500e-04, 8.0832376e-07, 1.3463484e-05, 1.6170477e-08,
        5.5103260e-06, 6.6362241e-05, 8.7910949e-04, 5.9447886e-04,
        8.1496772e-08, 2.6262711e-05, 5.2378509e-06, 1.7385283e-02,
        2.9627688e-04, 1.8729312e-05, 3.1481519e-05, 1.7897807e-04,
        2.9351559e-05, 6.0757752e-06, 1.5068624e-09, 1.1172429e-07,
        1.1447588e-09]], dtype=float32)
```

```
#Sentiment
model2.predict(instance)
```

```
array([[5.8543362e-04, 7.1172571e-01, 3.7886853e-06, 2.8768504e-01]],
      dtype=float32)
```

Figure 9.12: Output for User Instance Part 2

```
0      LAPTOP#OPERATION_PERFORMANCE
1                    LAPTOP#QUALITY
2                               NaN
3                               NaN
4                               NaN
                   ...
168         KEYBOARD#DESIGN_FEATURES
169                             NaN
170                     OS#USABILITY
171                     LAPTOP#PRICE
172                             NaN
Name: sentences/sentence/2/Opinions/Opinion/0/_category, Length: 173, dtype: object
```

```
print(labs_test)
```

```
[1 1 0 0 0 3 1 3 3 3 0 3 3 0 3 0 0 0 1 3 1 0 0 0 1 0 0 3 3 0 0 3 3 1 1 1 1
 1 0 1 0 3 0 3 0 0 0 1 3 0 3 0 0 3 3 0 0 3 0 0 3 3 0 0 0 1 3 3 3 1 0 0 1 0
 0 1 1 1 0 0 0 0 0 0 0 1 3 0 0 3 1 0 0 0 3 1 3 3 1 1 3 3 0 0 3 0 2 3 1 3 3
 3 3 1 1 1 3 0 1 3 3 3 0 0 0 0 0 0 0 3 0 0 3 1 1 0 1 0 3 0 0 0 0 3 1 0 1 1
 1 3 1 0 2 1 3 0 2 3 3 1 1 1 0 3 3 0 1 0 3 0 1 3 0]
```