

University of Sheffield

Sentiment Detection and Tracking in Social Media Streams



Sanchit Jitendra Ramteke

Supervisor: Dr. Mark Hepple

A report submitted in fulfilment of the requirements
for the degree of MSc in Advanced Computer Science

in the
Department of Computer Science

September 14, 2022

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Sanchit Jitendra Ramteke

Signature: S.J.R

Date: 14/09/2022

Abstract

Sentiment Analysis also called Opinion Mining is a type of text analysis that analyzes people's opinions about products, services, brands, and so forth. Sentiment analysis brings various research areas such as natural language processing, data mining, and text mining together and is fast becoming of great importance to businesses and organizations as it incorporates online commerce data for analysis purposes. Sentimental analysis can be done on different levels the focus of this research will be on document level sentiment analysis and Aspect-based sentiment analysis (ABSA). The first part of the experiment we focus on document level sentiment analysis comparing the classical approach for sentiment analysis and then showing the performance improvement observed when a state-of-the-art model like BERT is used for the same task. For the second part we move on the ABSA, which seeks to identify fine-grained opinion polarity regarding a particular aspect. For the purposes of this research, we construct auxiliary sentences from the aspect to provide information to the model on how the aspects are associated with the review. We use a pre-trained model like BERT and fine-tune it to achieve a high performance on Sentihood, SemEval 2014 Task 4 and SemEval 2016 Task 5 datasets.

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Mark Hepple, who has provided guidance and met with me face to face every week for the past three months. Apart from that, he provided me with a lot of critical advice and feedback when I was stuck on a task. His careful and patient guidance has been present throughout the entirety of the project, from its inception all the way through its completion.

In addition, I would like to express my gratitude and acknowledgement to my parent, who has not only loved and supported me without condition but also given me the opportunity to travel the world. Because of their support, I am able to persevere through many challenging times. I would not be able to accomplish this goal without their understanding and encouragement. I would also like to thank my friends for providing their support for the past three months.

Contents

1	Introduction	1
1.1	Aims and Objectives	1
1.2	Overview of the Report	2
2	Literature Survey	3
2.1	Sentiment Analysis (SA)	3
2.1.1	Different levels of sentiment analysis	4
2.1.2	Sentiment analysis tasks	4
2.2	Twitter Sentiment Analysis	5
2.2.1	Approaches for Twitter Sentiment Analysis	6
2.3	Aspect-based Sentiment Analysis (ABSA)	6
2.3.1	Approaches for ABSA	6
2.4	SemEval Tasks	7
2.4.1	Subtask A of SemEval 2017 task 4 - Twitter Sentiment Analysis . . .	7
2.4.2	SemEval ABSA Tasks	7
2.5	Lexicon based Techniques	8
2.6	Traditional Machine Learning Techniques	8
2.6.1	Naive Bayes	8
2.6.2	Support Vector Machine (SVM)	9
2.7	Deep Learning Techniques	11
2.7.1	Pre-trained Word Vectors	11
2.7.2	Convolutional Neural Network (CNN)	11
2.7.3	Recurrent Neural Network (RNN)	12
2.7.4	Long-Short Term Memory Neural Network (LSTM)	13
2.7.5	Transformer	14
2.8	Transfer Learning	17
2.9	BERT	18
2.9.1	BERT Architecture	18
2.9.2	Input/Output Representations for BERT	18
2.9.3	Sentence-pair classification task	19
2.9.4	BERT Pre-training task	19
2.10	Summary	20

3 Planned Experimentation	21
3.1 Aim and Objectives of the Project	21
3.2 Data	21
3.2.1 Twitter Sentiment Analysis	21
3.2.2 Aspect Based Sentiment Analysis (ABSA)	22
3.3 Methodology for the experimentation	22
3.4 Evaluation Metrics	22
3.5 Data Preprocessing	23
3.6 Feature Extraction	23
3.7 Programming Language	23
3.8 Relevant Libraries	24
3.8.1 NLTK	24
3.8.2 Scikit - Learn	24
3.8.3 PyTorch	24
3.9 Ethical, Professional and Legal Issues	24
4 Experiments and Results	25
4.1 Twitter Sentiment Analysis	25
4.1.1 Twitter dataset description	25
4.1.2 Baseline Experiment - Naive Bayes & SVM	26
4.1.3 Experiment I - Sentiment Analysis using BERT.	27
4.2 Aspect Based Sentiment Analysis (ABSA)	30
4.2.1 Experiment II - Targeted Aspect Based Sentiment Analysis (TABSA) on Sentihood Dataset	30
4.2.2 Experiment III - ABSA on SemEval-2014 Restaurant Dataset	32
4.2.3 Experiment IV - ABSA on SemEval-2016 Laptop Dataset	34
5 Conclusion and Future Work	37
5.0.1 Conclusion	37
5.0.2 Future Work	38
Appendices	43
A Supporting Images for the report	44
A.1 BERT Sentiment classification architecture	44
A.2 XML Snippet for SemEval-2014 Restaurant Dataset	44

List of Figures

2.1	SVM optimised hyperplane	10
2.2	CNN architecture with two channels with a sample sentence	11
2.3	RNN architecture	12
2.4	LSTM architecture [5]	14
2.5	Transformer architecture with Encoder and Decoder [39]	15
2.6	Transformer Attention Mechanism [39]	17
2.7	Input Representations for BERT[9]	19
4.1	Example of Tweets in SemEval-2017 dataset	26
4.2	BERT Pipeline for Sentiment Analysis	27
4.3	Example of Sentihood reviews with Auxiliary sentence	31
4.4	Example of Restaurant review with Auxiliary sentence	33
4.5	Example of Laptop review with Auxiliary sentence	35
A.1	BERT Pipeline for Sentiment Analysis	44
A.2	XML snippet for SemEval-2014 Restaurant Dataset	44

List of Tables

4.1	Polarity distribution - Twitter dataset	25
4.2	Results for Naive Bayes	26
4.3	Comparison with Baseline for Twitter Sentiment Analysis	29
4.4	Results for Twitter Sentiment Analysis	29
4.5	Example from Sentihood datset	30
4.6	Performance on Sentihood dataset	32
4.7	Polarity distribution SemEval-2014 Restaurant dataset	32
4.8	Performance on SemEval 2014 Restaurant dataset	33
4.9	SemEval-2016 Laptop dataset structure	34
4.10	Different BERT model version properties.	35
4.11	Performance on SemEval 2016 Laptop dataset	36

Chapter 1

Introduction

Sentiment analysis is the application of natural language processing to extract a feeling or emotion expressed in a piece of writing. Sentiment analysis is typically applied in fields such as data mining, web mining, and social media analytics because emotions are the most important features for evaluating human action and behaviour. The primary objective of opinion mining is to understand the viewpoint(s) of an individual or group regarding a target item by analyzing a large amount of text about the topic from multiple sources. Sentiment analysis is increasing used for understanding the purchasing intent of customers across variety of channels. It helps companies to identify customers who are ready to buy, making better predictions about when and where to move in a marketplace.

1.1 Aims and Objectives

The aim of this research work is to build a robust sentiment analysis system using Natural Language Processing (NLP) tools and advanced deep learning techniques. Also, exploring pre-trained models based on Transformer like Bidirectional Encoder Representations from Transformers (BERT) to compare the performance between various machine learning techniques. These aims will be trained and tested on gold standard datasets provided by SemEval conferences from multiple years and Sentihood dataset from [34]. We will be specifically considering datasets from SemEval conferences Twitter dataset from SemEval 2017 subtask A, ABSA datasets from SemEval 2014 Task 4 and SemEval 2016 Task 5. For the twitter dataset, the task is as follows given a tweet, decide whether the given tweet is positive, negative or neutral sentiment. For the Sentihood and SemEval 2016 Task 5 datasets we will seek to identify fine-grained opinion polarity regarding a particular aspect for a given review. We will also aim to improve pre-existing feature engineering techniques to improve the performance of the proposed model.

1.2 Overview of the Report

The research project is organized in the format given:

Chapter 2: Introduction of sentiment analysis and its various levels, the main tasks are also included. Then analyze the present situation and challenges of sentiment analysis in Twitter. The chapter then describes the feature engineering the datasets need to go thorough before they can be feed into the machine learning models, the machine learning approach and deep learning approach in detail. It will also go-through the literature review relating to ABSA. Finally, the specific tasks that we will perform in the project and the previous work that has been done relating to our work.

Chapter 3: First and foremost, a discussion of the aims and particular objectives of the project. Then, a description of data used for the experiments and the plan for the experiment, including the data pre-processing, evaluation metrics, programming language, feature selections, relevant libraries and ethical, professional, legal Issues.

Chapter 4: In this chapter, we go-over all the experiments performed for this report and their respective results. We also discuss about the proposed models and how they performed in their respective experiments.

Chapter 5: This chapter provides a concise overview of the entire research project by focusing on the objectives of the study. In conclusion, the significance of the findings of this research is discussed, along with potential follow-up studies that can be done following this research.

Chapter 2

Literature Survey

In this chapter, the literature on Twitter Sentiment Analysis is introduced and discussed thoroughly. Levels, tasks and approaches of sentiment analysis in Twitter are presented. We also go-through the necessity of aspect-based sentiment analysis, why this approach is more effective than previously mentioned sentiment analysis. Then the traditional machine learning and deep learning algorithms are introduced in detail. The chapter also discusses task, the subtask and data used from SemEval conferences. We end the chapter by discussing about how the pre-trained models can be utilized for Sentiment Analysis.

2.1 Sentiment Analysis (SA)

Sentiment analysis, also known as SA, refers to the process of automatically extracting sentiment information from text written in natural languages. The system or applications have been used for different purposes, for example, deciding the polarity of client reviews, following political opinion across social media streams, anticipating financial markets movement. With the rise of social media and ecommerce websites, people are increasingly comfortable expressing opinions about products and services online. People are used to sharing their thoughts and opinions online due to widespread use of the Internet. Meanwhile, with the help of internet, those expressions of sentiment and opinions are readily available to both individual researchers and organizations. There is real-world value in being able to extract sentiment expressions from the large data sets. For instance, reading reviews written by other customers can help to decide whether or not to buy a product, while reading reviews written by customers can give sellers new ideas on how to enhance their own offering. That being said, it is unrealistic to use manual input to analyse sentiment from the Internet because of the economic and time costs. Employing experienced experts to analyse all the texts is expensive and time-consuming. In addition, human beings are subject to cognitive and emotional biases that can cloud their judgement[21]. For the reasons above, it is imperative to use machines to analyse sentiment automatically and computationally[28].

2.1.1 Different levels of sentiment analysis

Prior studies have primarily focused on three levels of granularity when investigating sentiment analysis: the document level, the sentence level, and the aspect level[21][28].

Document Level : The goal of document-level sentiment classification is to figure out if an opinionated document that talks about an object has a positive or negative view of the object as a whole. For instance, a sentiment analysis system can tell the measure of how positive or negative a customer review about a certain product is as a whole. Since the results of sentiment analysis typically only have two (positive and negative) or three outputs(positive, negative and neutral), this level of sentiment classification presumes that a single document expresses opinions on a single object, such as customer reviews of products and services. Such an analysis relies on the assumption that the views expressed in each document directly relate to a singular topic (such as a specific product). This rule does not apply to document that compare or evaluate different entities[20].

Sentence Level : The task at this level is to determine the polarity of each sentence, i.e. positive, negative, and neutral sentiment. There are actually two distinct tasks here[17]. One of these is determining whether a given sentence is subjective or objective. A subjective sentence expresses the author's feelings or opinions about the subject matter, while an objective sentence presents facts. While this is generally true, there are exceptions; for example, "I bought a new car five days ago and the windshield wiper broke"[17] is an objective statement that also carries an emotional undertone. Sentiment classification, which entails labelling a sentence as positive, negative, or neutral, is another type of subtask.

Aspect Level : The task at this level, which is also known as the feature level, the objective is to identify sentiments regarding specific entities or/and their aspects. In contrast to the document level and the sentence level, the aspect level does not solely concentrate on linguistic constructions(phrases, sentences, paragraphs or document), however, the emphasis is placed more on specific aspects or characteristics[20]. For example, the user's comment may contain different aspects such as: "This book is a hardcover version, but the price is a bit high". The polarity of the 'appearance' is positive. and the polarity of the 'price' is negative[36]. Aspect level sentiment analysis aims to identify polarity towards a specific aspect. Users can get a more nuanced understanding of the quality of a product or service by performing this task and evaluating aggregated sentiments for each aspect of that product or service[36].

2.1.2 Sentiment analysis tasks

Bing Liu[21] devised a quintuple model to characterise the constituents of opinions for use in sentiment analysis tasks:

- e_i - the entity represented in the document.
- a_{ij} - unique aspect for the an entity e_i
- oo_{ijkl} - shows the sentiment associated with the aspect a_{ij}

- h_k - the opinion holder.
- t_l - the time when the opinion was expressed.

Find all the opinion quintuples as given above in a set of documents with opinions (D). This analysis can be summed up as follows from Liu[21]. Specifically, one must do the following tasks to attain this objective :

Task 1 (entity extraction and grouping): Extract all entity expressions in D, and group synonymous entity expressions into entity clusters. Each entity expression cluster indicates a unique entity e_i .

Task 2 (aspect extraction and grouping): Extract all aspect expressions of the entities, and group aspect expressions into clusters. Each aspect expression cluster of entity e_i indicates a unique aspect a_{ij} .

Task 3 (opinion holder and time extraction): Extract these pieces of information from the text or unstructured data.

Task 4 (aspect sentiment classification): Determine whether each opinion on an aspect is positive, negative or neutral. of information from the text or unstructured data.

Task 5 (opinion quintuple generation): Produce all opinion quintuples (e_i , a_{ij} , o_{ijkl} , h_k , t_l) expressed in D based on the results of the above tasks. A simple example on how to obtain the opinion quintuple is given below :

Given a review as document from User1 :

“We bought a Macbook Pro yesterday. The laptop is very fast but the camera is not good but its fine for normal video calls.”

After following all the tasks above we get the following:

- Task 1 - Macbook Pro
- Task 2 - laptop_processing, camera
- Task 3 - User1, 14-Aug-2022
- Task 4 - positive, negative
- Task 5 - See below.

The opinion quintuples generated for Task 5 are as follows :

- (Macbook pro, laptop_processing, positive, User1, 14-Aug-2022)
- (Macbook pro, camera, negative, User1, 14-Aug-2022)

2.2 Twitter Sentiment Analysis

The study of Twitter’s sentiments has become a popular topic of study recently. Twitter is a popular micro-blogging service because it allows users to quickly and easily share their thoughts and ideas with others[12]. Opinion mining and sentiment analysis can benefit from

the ever-expanding user base of micro-blogging platforms and services.

The following are some questions that may be of interest to manufacturers:

- How do customers feel about our (product, service, company, etc.)?
- How do people feel about our product, and is it generally well received?
- What would people like to see in our product?

The level of public approval could be useful information for political parties[27]. People on Twitter have posted a huge number of short messages they've written themselves. Ranging from personal opinions about a certain topic to broader political views. Because of these reasons, many have focused to do research on this topic. We will also focus first part of this report on Twitter Sentiment Analysis.

2.2.1 Approaches for Twitter Sentiment Analysis

To determine whether a tweet is positive, negative, or neutral is the task of sentiment analysis. For SemEval Task, Subtask A, we use the Semeval-2017 dataset[33]. This task is being run since 2013 and has been a recurring task in SemEval(2013[23], 2014[33], 2016b[24]). Recent studies show that Twitter posts use sentiment analysis (also called “sentiment classification”) in the following approaches[37]:

- Lexicon based
- Machine Learning based

All the mentioned approaches will be discussed in section 2.6, 2.7 and 2.8 in detail.

2.3 Aspect-based Sentiment Analysis (ABSA)

Hu[16], who pioneered the field of Aspect-based Sentiment Analysis (ABSA), argued that sentiment analysis can be studied on three different levels: the document level, the sentence level, and the entity or aspect level. The problem with focusing on the document or sentence level is that it assumes there is only one topic expressed. To fix this a more fine-grained approach is required where we are associating entity with an aspect and this pair of entity and aspect have their own specific sentiment associated with it. Over the past ten years, there has been a rise in interest in the field of Aspect-based sentiment analysis, which seeks to extract complex and subtle emotional expressions from text. We will be using a similar approach as discussed in section 2.3 for performing ABSA.

2.3.1 Approaches for ABSA

While an ABSA system is more difficult to design and implement than a traditional one, its practical benefits make the extra effort worthwhile. This system will allow us to handle

sentences with multiple entities with its associated aspect. Therefore, Saeidi[34] presented the task of targeted aspect-based sentiment analysis (TABSA), which seeks to determine fine-grained opinion polarity toward a particular aspect associated to the given target(target is the entity in our case).

In the prior research, there have been two strategies that have been bought up quite a bit that can be used to implement an ABSA system. The first method is rule-based or lexicon-based approach, and it can directly classify the polarity of the sentiment. The second strategy is the one in which supervised algorithms are put into practise. Many of the newer research use neural network-based methods[25, 40] which achieved higher accuracy than the standard models. Pre-trained models have recently shown their effectiveness to reduce the burden of feature engineering, examples include ELMo[29], Open GPT[7] and BERT[9].

If we consider BERT, it accomplished very successful results in Question Answering (QA) and Natural Language Inference (NLI). As mentioned in Sun[36], the utilization of pre-trained BERT model directly did not result in a significant improvement in the TABSA task. We have reason to believe that this may have been caused by an incorrect application of the pre-trained BERT.

In this report, we will be using the feature engineering methods mentioned by Sun[36] to get state-of-the-art results from our proposed model.

2.4 SemEval Tasks

The purpose of SemEval, an ongoing series of international natural language processing (NLP) research workshops, is to improve the state of the art in semantic analysis and to aid in the development of high-quality annotated datasets for a wide variety of increasingly difficult problems in natural language semantics[4]. The following describes the relevant SemEval tasks that the experiments are performed on.

2.4.1 Subtask A of SemEval 2017 task 4 - Twitter Sentiment Analysis

Task 4 of SemEval 2017[33] is focused on sentiment analysis in Twitter and is broken down into subtasks A, B, and C (ordinal tweet sentiment classification) and D and E (tweet quantification). Specifically in this report we will be focusing on subtask A, where we have to analyse a tweet and determine if it's positive, negative, or neutral sentiment.

2.4.2 SemEval ABSA Tasks

In order to evaluate ABSA, researchers at SemEval-2014[32] made available datasets of annotated reviews of restaurants and laptops. However, in 2014, researchers only had access to partial reviews in the datasets they used. However, comprehensive product evaluations were not included until SemEval-2015[31]. With the exception of some additional test data, the datasets were kept the same from SemEval-2014[32] until SemEval-2016[30]. The SemEval ABSA task aimed to identify expert and user perspectives on a given topic from customer

reviews. The SemEval-2016 was divided in to three slots, for the purposes of this report we will focusing only on the Slot 3 which is Sentiment Polarity classification of reviews associated with its specific aspects. We will be designing a method for categorising reviews based on the positive, negative, or neutral sentiment expressed about a given feature.

2.5 Lexicon based Techniques

Calculating a document's orientation based on the semantic relationships between words and phrases contained in the document is the focus of the lexicon-based technique[38]. Constructing classifiers from labelled examples of texts or sentences is at the centre of the text classification method[28], which means that this classification process is basically a supervised one. There are two ways in which lexicon based sentimental analysis can be carried out the first is dictionary-based and the second is corpus-based. The dictionary is the centre of the dictionary-based method, which uses it to find related words to the initial seed words. To retrieve semantic orientation, the corpus-based method begins with a list of opinion words and then finds out additional opinion words in the large corpus.

2.6 Traditional Machine Learning Techniques

Many of the traditional machine learning techniques are supervised learning algorithms. For all the supervised learning algorithms, there must be labelled training data for these techniques to work and provide successful results. The term “supervised” is intended to make the user think of a supervisor who provides guidance to the learning system which labels to relate/link with the training examples. In general classification problems, these labels typically identify different classes for the purposes of this report we are classifying the reviews into three sentiment classes positive, negative and neutral. The following section provides an overview of the classifiers used for this research and some of the popular classifiers in sentiment analysis.

2.6.1 Naive Bayes

When it comes to machine learning and data mining, Naive Bayes is one of the most effective and efficient inductive learning algorithms available. The Naive Bayes is a type of conditional probability model[22]. Given a classification prediction problem, the conditional probability $P(y|x)$ formula can be given as follows :

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (2.1)$$

- $P(y)$ is the “prior probability” - training data label probabilities
- $P(x)$ is the “evidence” - the probabilities of features in training data

- $P(x|y)$ is “likelihood” - how likely it is that someone belonging to class y has feature x
- $P(y|x)$ is the “posterior probability” - how likely it is that someone belonging to feature x has class y

To proceed, we make the “naive” conditional independence assumptions, which state that given a category, all features in are mutually independent. Therefore, if we assume that

$$p(x_i|y) = p(x_1, x_2, \dots, x_n|y) = \prod_{i=1}^n p(x_i|y) \quad (2.2)$$

Using the aforementioned equations to develop a classifier

The maximum a posteriori (MAP) decision rule recommends selecting the more likely hypothesis in order to reduce the risk of incorrect classification. A Bayes classifier is the corresponding classifier, and it is the function that assigns a class label $\hat{y} = C_k$ for some k :

$$\hat{y} = \operatorname{argmax}_{k \in \{1, \dots, K\}} p(C_k) \prod_{i=1}^n p(x_i | C_k). \quad (2.3)$$

When using a Naive Bayes network, all attributes are considered independent no matter the value of the class variable. Conditional independence describes this situation[42].

2.6.2 Support Vector Machine (SVM)

Corpus-based supervised learning is now standard for natural language processing however supervised learning requires a huge annotated corpus. Without a sufficiently large annotated corpus, even a good supervised algorithm won’t work. Annotating corpora is time-consuming and requires a large amount of capital. This problem is solved by SVMs as it creates virtual examples based on what it already knows about the task, increasing the dataset size on which our model will train. The below section defines SVMs theoretically.

Assume training data[35] :

$$(x_i, y_i), \dots, (x_l, y_l), x_i \in R^n, y_i \in \{+1, -1\} \quad (2.4)$$

In SVM, decision function g is defined as ,

$$g(x) = \operatorname{sgn}(f(x)) \quad (2.5)$$

$$f(x) = \sum_{i=1}^l y_i \alpha_i K_i(x_i, x) + b \quad (2.6)$$

where K is the kernel function, K is defined as $K(x_i, x) = x_i \cdot x$.

$b \in R$ is a threshold, and α_i are weights. α_i should satisfy the following constraints :

$$\forall i : 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^l y_i \alpha_i = 0$$

Now, equation (2.6) can be written as follows :

$$f(x) = w \cdot x + b \quad (2.7)$$

where $w = \sum_{i=1}^l y_i \alpha_i K(x_i, x)$. To train an SVM is to find α_i and b by solving the following:

$$\text{maximize : } \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (2.8)$$

$$\text{depending on : } \forall i : 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^l y_i \alpha_i = 0 \quad (2.9)$$

The solution provides a hyperplane that acts as a cutoff between the two classes[35]. The Figure 2.1 below shows the optimised hyperplane and support vectors.

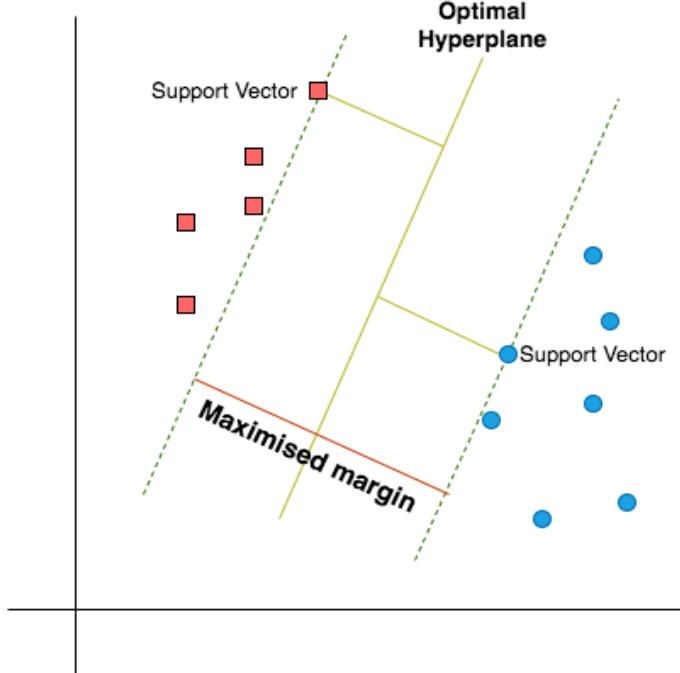


Figure 2.1: SVM optimised hyperplane

2.7 Deep Learning Techniques

Many of the Deep learning techniques are quite successful in achieving exceptional efficiency in a wide range of natural language processing tasks such as for text classification. We will be using same techniques for the purposes of our research. We will discussing about unsupervised learning where the training set in unsupervised learning does not contain any labels. Most of the time, a network's effectiveness is measured by whether or not it is able to decrease a cost function[26]. Some of the most popular techniques are discussed in the following section.

2.7.1 Pre-trained Word Vectors

When building neural network models for natural language processing tasks like sequence tagging and text classification, pre-trained word embeddings have proven to be very useful. Pre-trained word vectors like GloVe and Word2Vec can transform words into useful vectors using continuous vector representations of words. Recently, many of the researchers have used word pre-trained word vectors to improve their proposed models performance.

2.7.2 Convolutional Neural Network (CNN)

CNNs use layers with convolving filters on local features[19]. CNN models were originally developed for Computer Vision(CV) but have proven effective for NLP tasks such as semantic parsing, search query retrieval, sentence modelling, and more[8]. Figure 2.2 shows a variant of Collobert[19] architecture. The Convolution Neural Network comprises a series of filters

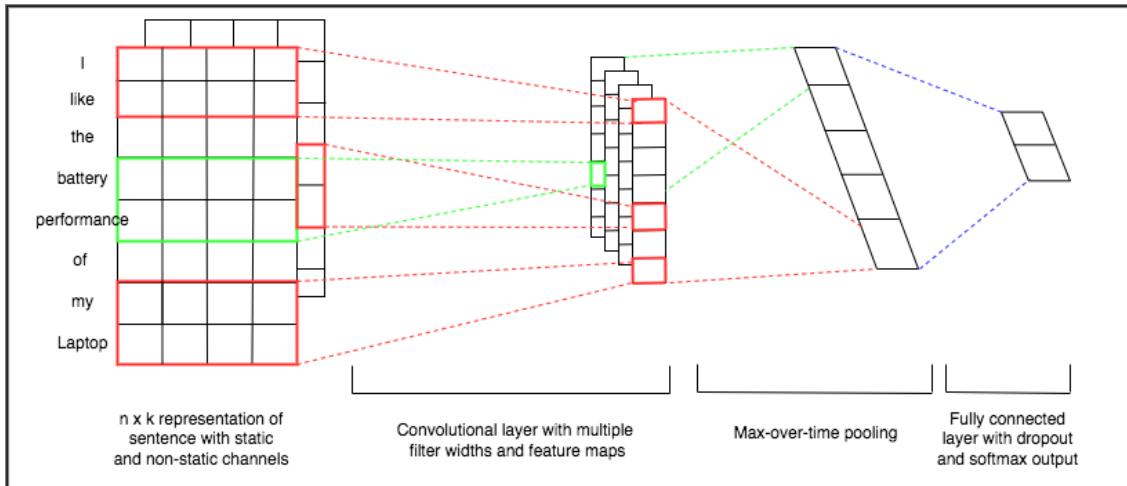


Figure 2.2: CNN architecture with two channels with a sample sentence

of varying dimensions and shapes. These filters convolve, or roll over, the original sentence matrix in order to reduce it into additional low dimension matrices. CNNs are currently being applied to distributed and discrete word embedding in the field of text classification[10].

As we discuss before CNN can be used for NLP tasks but the use of CNN is more appropriate for CV tasks than Natural Language Processing ones. The input data of an

image for a straightforward CV mission is represented by individual pixels that correspond to various colours and tonalities. The network learns without regard to the sequence of the input images because each image is processed individually. However, when learning textual data, it needs to be treated as a series of words rather than processing the words independently of each other in order to capture details such as the meaning of the words and the semantic context of the words in the sentences. This issue is solved by the next model(RNN) we will be discussing in section 2.7.3.

2.7.3 Recurrent Neural Network (RNN)

Traditional ways of representing features often don't take into account the context or the order of the words in a text, so they don't do a good job of capturing the semantics or meaning of the words. For example, we take a look at the following sentence "*I like the battery performance of my Laptop.*". If we consider the word *battery*(unigram) we don't know whether it means the cells we put in electrical appliances or the act of beating someone. Once we have the larger context "*battery performance of the Laptop*"(5-grams), it becomes simple to differentiate the meaning. Recurrent Neural Network solves this issue. This model takes a sentence or paragraph at a time, dissecting it word by word before storing the hidden layer's semantic weights in a single, predetermined-size hidden layer[11]. Compared to other methods, RNN is superior in its ability to pick up on context. Semantic information from lengthy texts is more easily captured with this method[18]. RNN architecture is shown in Figure 2.3.

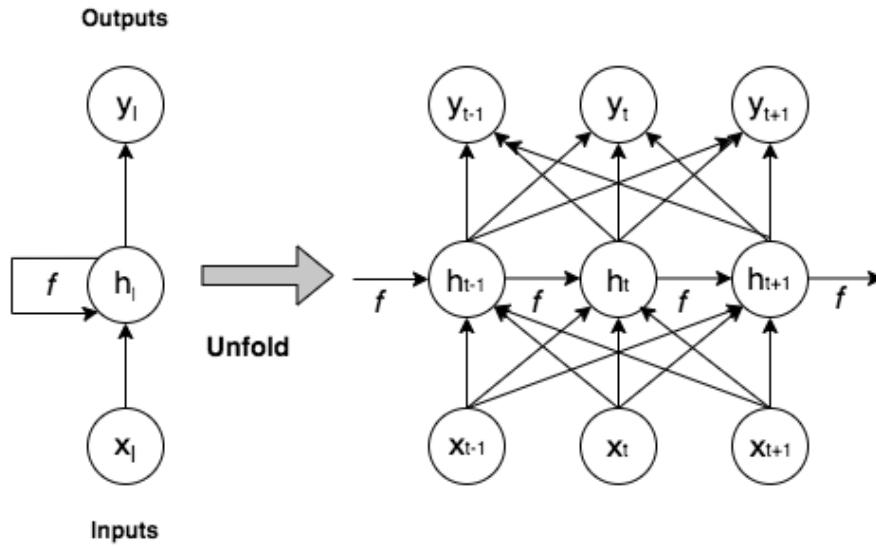


Figure 2.3: RNN architecture

The time series which is provided as input to the model is given as follows $X = (x_1, x_2, \dots, x_n)$, the hidden state sequence is as $H = (h_1, h_2, \dots, h_y)$ and the sequence

obtained at the output layer $Y = (y_1, y_2, \dots, y_n)$. We can see from the following equations show how X , H , and Y are related to one another[11]:

$$h_n = \sigma(W_{xh}x_n + W_{hh}h_{n-1} + b_h) \quad (2.10)$$

$$y_n = W_{hy}h_n + b_y \quad (2.11)$$

where,

- σ is the non-linear activation function
- W_{xh} weight matrix from input to hidden layer
- W_{hh} weight matrix from hidden layer to hidden layer
- W_{hy} the weight matrix from hidden layer to output
- b_h and b_y are biased terms

2.7.4 Long-Short Term Memory Neural Network (LSTM)

It take a long time to learn how to store information over prolonged time intervals using recurrent backpropagation. This is primarily because there is inadequate error back flow that is decaying over time [15]. To address the problem of inadequate error back flow LSTM architecture was introduced. The Long short-term memory (LSTM) neural network model is a variation on the RNN architecture regularly utilized for addressing sequence problems. The LSTM makes it possible to keep the data extractions for a long time as the same network and weight parameters are used in every timestep. The LSTM architecture is shown in the figure below. The three control units, or gates, of the LSTM proposed[15] are the input gate i_t , the forget gate f_t , and the output gate o_t , and they work together to maintain the memory C_t at time t , as depicted in Figure 2.4 and described in the formulas as follow:

$$\begin{aligned} f_t &= \sigma(W_{hf}h_t + W_{cf}c_{t-1} + W_{xf}x_t + b_f) \\ i_t &= \sigma(W_{hi}h_t + W_{xi}c_{t-1} + W_{ci}x_t + b_i) \\ o_t &= \sigma(W_{oh}h_{t-1} + W_{xo}x_t + b_o) \\ \tilde{c}_t &= \tanh(W_{hc}h_t + W_{xc}x_t + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \quad (2.12)$$

where,

- σ represents the element-wise sigmoid function,
- \odot denotes the element-wise dot product operator
- x_t is the input vector at time t

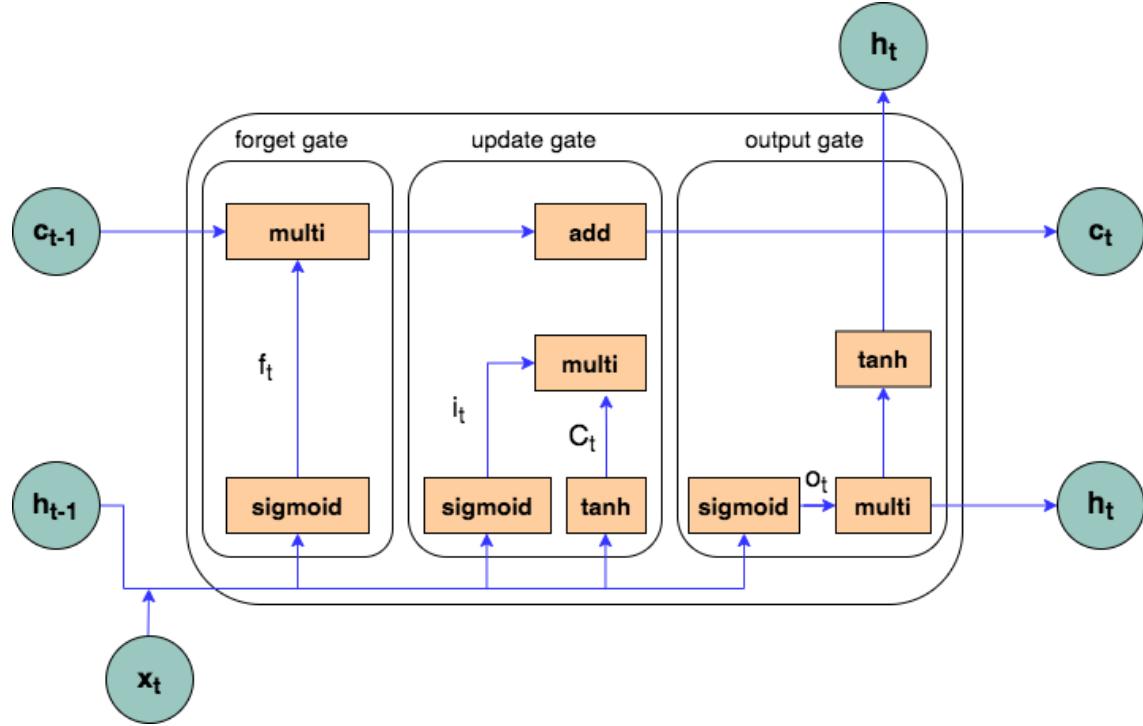


Figure 2.4: LSTM architecture [5]

- h_{t-1} is the hidden state vector that has all the relevant data stored ahead of time
- W_{xi}, W_{xf}, W_{xc} and W_{xo} show the input x_t weight matrices for various gates.
- W_{ci}, W_{cf} represents the weight matrices for cell state c_{t-1} .
- b_f, b_i, b_o and b_c represents the bias vectors.

2.7.5 Transformer

The Transformer[39] has rapidly become the dominant architecture for natural language processing, surpassing alternative neural models such as convolutional and recurrent neural networks in performance for tasks in both natural language understanding and natural language generation. The architecture scales with training data and model size, facilitates efficient parallel training, and captures long-range sequence features. The Figure 2.4 shows the Transformer architecture.

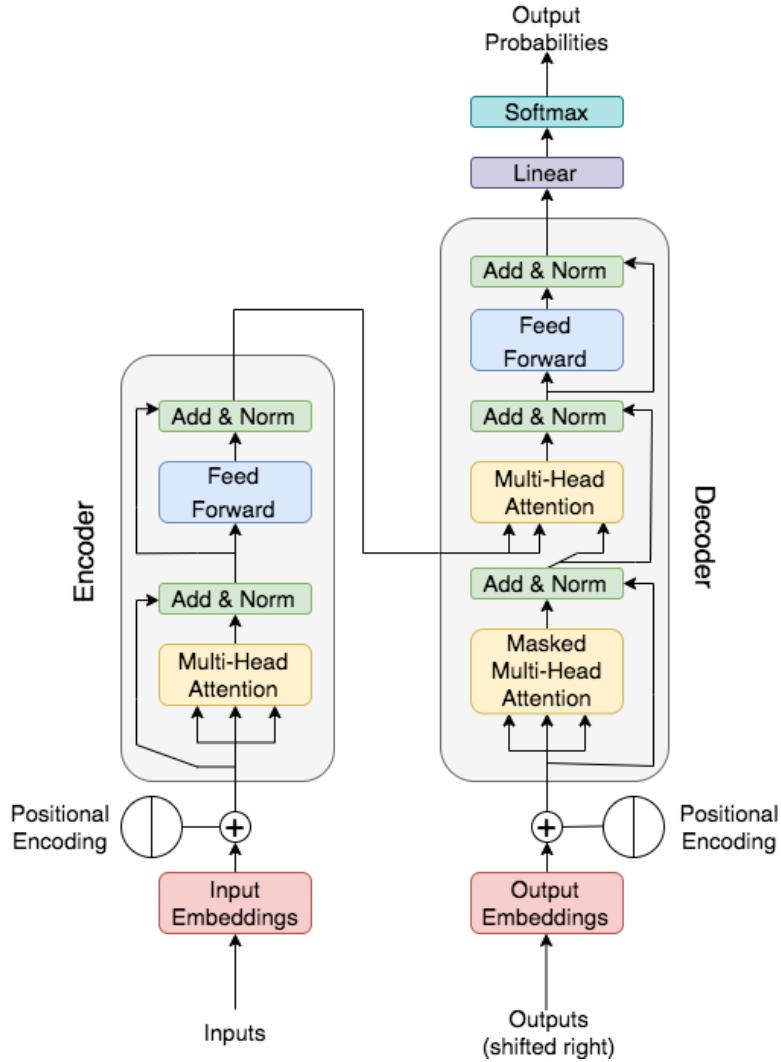


Figure 2.5: Transformer architecture with Encoder and Decoder [39]

In 2017, Vaswani[39] proposed a new Transformer design that makes use of multi-headed self-attention. Most cutting-edge NLP systems previously relied on gated RNNs like LSTMs and gated recurrent units (GRUs) with additional attention mechanisms before the advent of transformers. Although they lack the recurrent structure of RNNs, transformers also use attention mechanisms. Therefore, attention mechanisms alone can effectively compete with performance on RNN given enough training data[39].

Transformer Architecture

The Transformer architecture works by employing stacked self-attention and point-wise, fully connected layers for both the encoder and the decoder, as shown in the left and right halves of Figure 2.6, respectively. We will discuss the encoder and decoder in detail in the following

sections.

Encoder The encoder is made up of encoding layers, which process the input in an iterative manner, moving from one layer to the next. Encoder has 6 layers. Sublayers are on each layer. First is a multi-head self-attention mechanism, and second is a simple feed-forward network[39].

Decoder The decoder is made up of different layers of decoding the processing is same as that of the encoder. The output from the encoder is processed in an iterative manner by the decoder, which moves from one layer to the next. The decoder has 6 layers. The decoder adds a third sublayer to each encoder layer to perform multi-head attention on the encoder's output. As with the encoder, we use residual connections and layer normalisation. We also tweak the decoder stack's self-attention sub-layer to stop positions from paying attention to what comes next[39].

Attention An attention function maps a query and a set of key-value pairs to an output. The query, the key-value pairs, the output, and the output itself are all vectors. The output is calculated as a weighted sum of the values, where the weight given to each value is based on how well the query matches the key[39]. Two types of attention are employed by the transformer architecture:

1. Scaled dot-product attention : We figure out the attention function for a group of queries at the same time into a matrix Q . Additionally, the keys and values are packed into matrices K and V . The Q , K and V can be defined as follows :

- Query (Q): vector from which the attention is looking.
- Key (K): vector at which the query looks to establish context.
- Value (V): value of word being looked at, weighted based on context

The Scaled dot-product attention is computed as follows[39]

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.13)$$

The input consists of queries and keys of dimension d_k , and values of dimension d_v . The dot products of the query with all keys, divide each by $\sqrt{d_k}$, and to obtain the weights based on the values, we use a softmax function.

2. Multi-head attention : Multi-head attention relies on Scaled dot-product attention for its internal workings. The model is able to jointly attend to information coming from a variety of representation sub-spaces at a variety of positions thanks to the multi-head attention capability. Averaging prevents this from happening when there is only one

attention head. The formula for multi-head attention is[39]

$$\text{MultiHead}(Q, K, V) = [\text{head}_1, \dots, \text{head}_h] W_0 \quad (2.14)$$

where $\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$

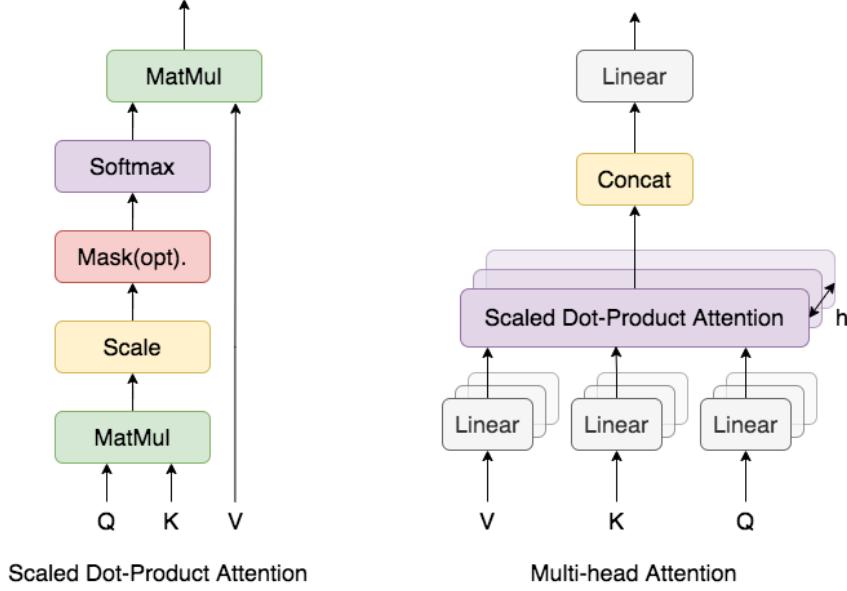


Figure 2.6: Transformer Attention Mechanism [39]

2.8 Transfer Learning

Transfer learning is a technique for machine learning in which a model created for one task is used as the basis for a model on another task. Humans can learn to solve new problems with very few samples, whereas training a model from scratch requires a large amount of data. The ability to apply previously acquired knowledge to solve novel problems drives this remarkable learning process. Motivated by this, transfer learning codifies a two-stage learning framework: a pre-training phase to extract information about one or more source tasks, and a post-training phase to apply that information to one or more target tasks. As a result of all the data collected during the pre-training phase, models can be fine-tuned to perform well on target tasks with fewer data points[13].

In transfer learning, two pre-training methods, feature transfer and parameter transfer, receive a lot of attention. Knowledge can be encoded in advance and applied to new domains and tasks with the help of feature transfer techniques. The performance of models on target tasks can be greatly improved by injecting these pre-trained representations. Methods for transferring parameters are based on the commonsense assumption that two tasks can share

the same model parameters or prior distributions of hyper-parameters. Consequently, these strategies transfer expertise by fine-tuning pre-trained parameters with the data of target tasks[13].

2.9 BERT

Bidirectional Encoder Representations from Transformers, or BERT for short, is an abbreviation. This allows the model to be used to create state-of-the-art models for a wide range of tasks[9]. Current techniques limit pre-trained representations' power, especially for fine-tuning. Standard language models are unidirectional, limiting pre-training architecture choices. In OpenAI GPT, every token can only attend to previous tokens in the Transformer's self-attention layers[39]. Such restrictions are not efficient for sentence-level tasks and could be harmful when applying finetuning-based approaches.

Previous models flaws led to the creation of BERT. BERT was developed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context across all layers. This is accomplished through the use of directional conditioning. Therefore, the pre-trained BERT model can be fine-tuned with only one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications[9].

2.9.1 BERT Architecture

Based on the initial implementation that was described in Vaswani[39], the BERT model architecture is a multi-layer bidirectional Transformer encoder. BERT models have significant number of encoding layers, which are referred as "Transformer Blocks" in the paper[9]. The base model which is the one we will be using in this report has twelve of these layers. This model also have a larger feed-forward network with 768 hidden units. Also, more attention head(12) than the default configuration of the original transformer which had 6 encoding layers, 512 hidden units and 8 attention heads.

2.9.2 Input/Output Representations for BERT

A process referred to as wordpiece tokenization is first applied to the text that is provided to the BERT model for processing [41]. This results in a set of tokens, where each one stands for a different word. In addition to the standard tokens, two specialised tokens are introduced.

- [CLS] - classifier token, added in the beginning of a sentence.
- [SEP] - separator token, added at the end of the sentence marking the end of the sentence.

Token embedding layer, segment embedding layer, and position embedding layer are all used to process this set of tokens before they are combined and sent on to the encoder layer. Token positions in a sequence are encoded using these embeddings. Figure 2.7 shows an

example of all the embedding layers for a sample sentence.

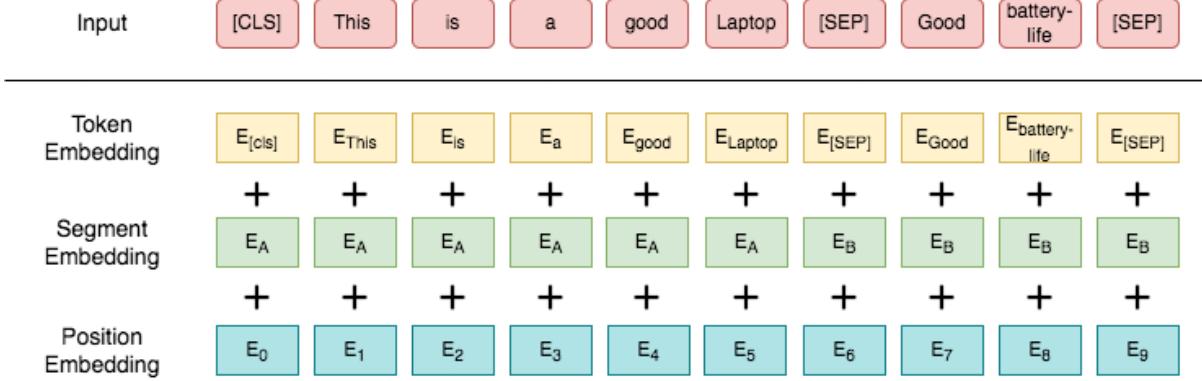


Figure 2.7: Input Representations for BERT[9]

2.9.3 Sentence-pair classification task

When working on the Sentence Pair Classifier task, this task deals with identifying semantic relationship between two sentences. The inputs for this task are two texts, and the model returns a label that describes the relationship between the sentences. In our Experiments we will be making application of this task due to BERT’s flexible input representation, we can easily transform ABSA into a sentence-pair classification task and perform fine-tuning on the pre-trained BERT. This gives us a better model performance than performing single sentence(document level) classification.

2.9.4 BERT Pre-training task

Masked Language Model To pre-train deep bidirectional representations for the language model, BERT makes use of a mask token [MASK]. BERT disguises a random word in the text sequence, as opposed to conditional language models that train left-to-right or right-to-left to predict words, with the predicted word located at the end or the beginning of the text sequence[14]. The other motivation for pre-training with a mask token is that standard conditional language models can only train left-to-right or right-to-left due to the directional indentations of the words[14].

Next Sentence Prediction The purpose of Next Sentence Prediction is to understand the connection between two sentences in a text. The BERT system has been trained to determine if there is a connection between two sentences. An example is shown below.

Sentence A : [CLS] I bought this Laptop . [SEP]
 Sentence B : Its battery life is great ! [SEP]
 Label : IsNextSentence

Embedding dimensions are unique to each of these sentences, A and B. In training, the IsNextSentence label is used when ‘Sentence B’ is the logical continuation of ‘Sentence A’ 50% of the time. Half the time, a predetermined sentence is used, but the other times, it’s chosen at random and the IsNotNextSentence label is used.

Data used for Pre-training For the pre-training corpus data used for training is the BooksCorpus (800M words) and English Wikipedia (2,500M words).

2.10 Summary

In this section, details on both Twitter based sentiment analysis and aspect based sentiment analysis are discussed in detail. We also go over various machine learning and deep learning techniques that are going to be utilized for performing experiments. In addition to this, we discussed a variety of SemEval tasks that we will be performing for this research. This chapter comes to a close with a discussion on transfer learning and the pre-trained model of BERT, both of which will be utilised in this report.

Chapter 3

Planned Experimentation

In this chapter, the main aims and objectives in this chapter. We also compare and contrast constrained and unconstrained systems to design a system that is capable of delivering on the project's needs.

3.1 Aim and Objectives of the Project

The main aims of this project is to design and implement a system that can accurately identify the sentiment polarity(negative, positive or neutral) of the given document. For the first part we will be focusing on conducting sentimental analysis on a document level using the gold standard twitter dataset. We will be comparing performance of various models to our proposed system. In the next part, we move on to performing Aspect based sentiment analysis which aims to identify fine-grained polarity towards a specific aspect. We will also be comparing performance of various models that have been provided as benchmark models in various previous research papers to our system.

3.2 Data

As we have discussed in section 2.4.2, we will be using Sentihood dataset from Saeidi[34] and datasets from various SemEval competitions. All the datasets we will be using are Gold Standard. Tasks and Datasets used for the experiments are given below.

3.2.1 Twitter Sentiment Analysis

This work will make use of information collected for SemEval 2017 task 4, subtask A (Sentiment Analysis in Twitter). In total, there are 50333 pieces of training data and 12284 pieces of test data. Each tweet can be placed in one of three categories: positive, negative, or neutral. The goal is to accurately categorise test data.

3.2.2 Aspect Based Sentiment Analysis (ABSA)

- Sentihood dataset - We will performing Target-Aspect based sentiment analysis (T)ABSA on Sentihood dataset[34] which is comprised of 5,215 sentences, of which 3,862 have a single target, and the remaining sentences have multiple targets. The Sentihood dataset is a real-estate reviews dataset, the locations mentioned in the reviews are the targets in this case.
- SemEval datasets - We will be using Restaurant dataset from SemEval-2014 and Laptop dataset from SemEval-2016.

For the ABSA experiments, we will be utilising a total of three datasets. The goal of this experiment is to identify correct polarity towards a specific aspect.

3.3 Methodology for the experimentation

The methods used to develop the sentiment classification system, we will be using both traditional machine learning techniques and deep learning techniques. Naive Bayes based classification system will be implemented first and considered as a baseline for Twitter sentiment analysis. In the second part, we will be using a combination of deep learning techniques(neural networks) and pre-trained models(BERT) to implement the proposed system. This will be used for both Twitter sentiment analysis and ABSA.

3.4 Evaluation Metrics

In order to compare the results with various research paper from the results obtained for the proposed system we will be using *Accuracy* as the primary evaluation metric. Accuracy can be computed as

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (3.1)$$

Where TP , TN , FP , and FN stand for True Positives, True Negatives, False Positives and False Negatives respectively.

For Twitter sentiment analysis we will also us the *Average Recall* metric which is provided in SemEval-2017[33]. The formula 3.2 for *Average Recall* is as shown below .

$$\text{AverageRecall} = \frac{1}{3}(R_P + R_N + R_U) \quad (3.2)$$

where R_P , R_N and R_U reference to recall, the following refers to POSITIVE, the NEGATIVE, and the NEUTRAL class, respectively. We also use *Macro F1 score* which is he unweighted mean of the F1 scores that were calculated for each class makes up the *Macro F1 score*. It is the aggregation for the F1 score that is the least complicated.The formula is as follows :

$$\text{Macro F1 score} = \frac{\text{sum}(F1 \text{ scores})}{\text{number of classes}} \quad (3.3)$$

where F1 scores are the normal F1 score for each class.

For ABSA we use the benchmark results from Sun[36] and compare results obtained from proposed model with models submitted for various SemEval conferences.

3.5 Data Preprocessing

As such, data preprocessing plays a crucial role in text classification models. We need to properly clean the data from unwanted stuff that might hinder the final results obtained after the models is implemented. As a result, we will need to get rid of extra words, punctuation, and other characters that don't really add anything to the meaning of the sentence. The following data preprocessing is done for Twitter dataset.

- Lowercase the whole sentence.
- Remove `@name`.
- Remove all stopwords such as ‘a’, ‘an’ as these don’t really have an impact in determining the polarity of a sentence except ‘not’ and ‘can’ as removing a ‘not’ from a sentence can have significant impact on classifying the polarity.
- Isolate and remove punctuation except ‘?’.
- Remove some special characted such as ([;,:—,•]). These characters don’t contribute in sentiment classification.
- Remove trailing whitespaces.
- Convert the text into tokens for further processing.

3.6 Feature Extraction

We will be using TF-IDF(Term Frequency - Inverted Document Frequency) for feature extraction as by multiplying tf(term frequency) and idf(inverted document frequency), TF-IDF is able to accurately capture words that are less common across the entire dataset but more common in a subset of data.

3.7 Programming Language

Because of its wide usage in the field of natural language processing, Python was selected as the language to be used in the implementation of this project. In addition to this, there are many libraries developed and made public such as Scikit-Learn, PyTorch which are essential for development of the proposed model.

3.8 Relevant Libraries

3.8.1 NLTK

NLTK[6] also known as Natural Language Toolkit will be implemented for the purposes of data-preprocessing, lemmatization and tokenization. It's a very easy to use tool for NLP.

3.8.2 Scikit - Learn

Scikit-Learn[2] is a free machine learning library written in Python. It is compatible with the Python numerical and scientific library NumPy and includes a number of classification, regression, and clustering algorithms. We will be implementing Naive-bayes and Support Vector Machine(SVM) from this library.

3.8.3 PyTorch

An open source machine learning framework, PyTorch[3] is built on the Torch library. Many of the major deep learning algorithms are built on PyTorch. HuggingFace provides all its libraries and pre-trained models like BERT in PyTorch, so for our experiments we have used PyTorch to implement the pre-trained models.

3.9 Ethical, Professional and Legal Issues

In this undertaking, no humans were involved in the data collection process. This project relied on a training and testing dataset that was made publicly available by the SemEval organisers and was made available through their website. The same goes for the Sentihood Dataset[34]. All third-party resources cited in subsection 3.8 are freely accessible online. They are available for use by any interested party for scientific investigation. There are no ethical or legal concerns with this undertaking for the reasons given.

Chapter 4

Experiments and Results

This chapter provides an in-depth analysis and description of the data. In addition, we discuss the results of all of the experiments that were conducted and go over the methodology that was utilised for all of the experiments.

4.1 Twitter Sentiment Analysis

In this section we discuss all the experiments done for performing Sentiment Analysis on Twitter dataset. We begin by discussing the experiment that was done to establish a baseline for this using traditional machine learning methods. After that, we move on to improving the performance of the model by utilising deep learning techniques and pre-trained models such as BERT.

4.1.1 Twitter dataset description

The Twitter dataset is a general dataset i.e., it does contain tweets for a particular domain. Figure 2.12 displays an example taken from the training data, with the word “sentiment” appearing in the first column and “tweet” appearing in the second column. The total number of training data instances and the number of test data instances; the distribution of these numbers is shown in Table below.

Dataset	Positive	Neutral	Negative	Total
Train	19,902	22,591	7,840	50,333
Test	2375	5,937	3,972	12284

Table 4.1: Polarity distribution - Twitter dataset

The data snippet in Figure 4.1 shows a few examples in Twitter dataset.

		tweets	labels
0	Ben Smith / Smith (concussion) remains out of ...		1
1	Sorry bout the stream last night I crashed out...		1
2	Chase Headley's RBI double in the 8th inning o...		1
3	@user Alciato: Bee will invest 150 million in ...		2
4	@user LIT MY MUM 'Kerry the louboutins I wonder...		2

Figure 4.1: Example of Tweets in SemEval-2017 dataset

4.1.2 Baseline Experiment - Naive Bayes & SVM

For the baseline method, we first do data preprocessing on the dataset as discussed in section 3.5. Then we perform feature extraction and vectorize our text data using TF-IDF. We have set the ngram range for this experiment as unigram. After that we will employ a Naive Bayes and SVM classifiers. We will be taking these two examples as the baseline scores to compare with the proposed model.

Results

The results obtained after performing sentiment analysis using Naive Bayes is given in the Table 4.2.

Experiment	Accuracy	F1 Score
Naive Bayes	54.42%	48.34%
SVM	58.65%	58.65%

Table 4.2: Results for Naive Bayes

As we can observe from the table 4.2 the reported accuracy for Naive Bayes is 52.49% and the macro averaged F1-score is 39.35%. Similarly, for SVM the reported accuracy is 58.65% and the macro averaged F1-score is 58.65%.

Discussion

Naive Bayes method gives the poor results for two factors. One is that it assumes each feature is independent, therefore when figuring out the prior probability, the association between features is not taken into account. Another factor is that it doesn't need to learn parameters to make predictions. Instead, it uses the statistical results of the information directly. As we can see from the results SVM performed 4% better than Naive Bayes, so we will be considering SVM as the standard baseline for the experiments ahead. One point to be noted here is the training time it takes for both the models, Naive Bayes takes significantly less time as compared to SVM on same amount of training data.

On account of ngram range, the experiments were performed on unigram, bigrams and trigrams. We obtained the best results on unigram on both Naive Bayes and SVM models. Also, the resources and time required to train the models with bigrams and trigrams increased significantly. But did not improve the model performance as expected.

4.1.3 Experiment I - Sentiment Analysis using BERT.

We will now use BERT and compare the results with Baseline and with the results obtained during SemEval - 2017[33].

Methodology

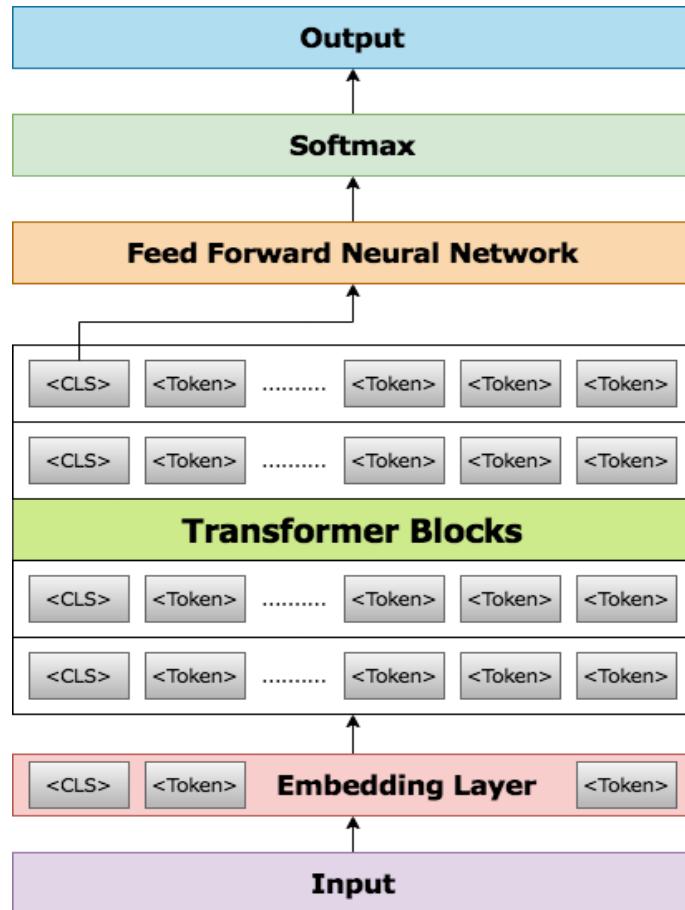


Figure 4.2: BERT Pipeline for Sentiment Analysis

Given that they are trained on the general language rather than a specific domain, the standard versions of the BERT model provide general language representation. This allows the model to be trained and improved for the downstream tasks with the available data. Since the model offers a set of output layers, it can be fine-tuned to carry out any of the NLP

tasks for which it is a candidate. During this stage, we'll fine tune in on the specifics of our data and our downstream task by tuning all of the model's parameters simultaneously.

The BERT model is able to support a wide variety of downstream tasks. Some examples of these tasks include text summarizing, question and answering and text classification. BERT is utilized for this study for the purpose of Sentiment Analysis which is basically a text classification task.

We will be using HuggingFace's transformer library[1] which uses PyTorch to implement models like BERT. This library's tokenizer is required for use with the pre-trained BERT. This is due to

1. the model has a predetermined, fixed vocabulary and
2. the BERT tokenizer handles words that are not in the model's vocabulary in a specific way.

We will use the *encode_plus* method from this library which will

1. separate our text into individual tokens.
2. add the special token [CLS] and [SEP] to the text.
3. rearrange these tokens so that the tokenizer can index them in its vocabulary.
4. adjust the length of sentences by padding or truncating as necessary.
5. attention mask creation

The *BertForSequenceClassification* class in the transformers library is tailor-made for such applications. We will, however, make a new category in order to select our own classifiers. Our classifier will be a single hidden layer feed forward neural network, and we will use a BERT model to extract the [CLS] token's final hidden layer the process has been illustrated in Figure 4.2. We need to make an optimizer in order to fine-tune our Bert Classifier. The authors[9] suggest the hyper-parameters below:

- Batch size : 16 or 32
- Learning rate : 5e-5, 3e-5 or 2e-5
- Epochs for training: 2, 3, 4

The hyper-parameters selected for our experiments and the ones which achieved the best results are :

- Batch size : 64
- Learning rate : 3e-5
- Epochs for training: 2

We will be using AdamW optimizer as suggested in examples provided by HuggingFace's BERT scripts.

Results

In this section we discuss the results obtained after performing the previously mentioned experiment using BERT.

Experiment	Accuracy	F1 Score
SVM(baseline)	58.65%	58.65%
BERT	71.35%	71.05%

Table 4.3: Comparison with Baseline for Twitter Sentiment Analysis

The above table 4.3 shows the Accuracy and the F1-Score for Twitter sentiment analysis done using BERT based model. Our proposed model performed much better than the baseline. We achieved an performance improvement of 12.7% as compared to the baseline. We will now compare the results obtained with the top two models that were presented in SemEval-2017[33].

- BB-twtr : BB twtr used an ensemble model with LSTMs and CNNs with multiple convolution operations
- DataStories : DataStories used deep LSTM networks with an attention mechanism[33]

The table 4.4 shows the performance improvement due to usage of BERT.

Experiment	Average Recall	Accuracy	F1 Score
BB-twtr(CNN+LSTM)[33]	68.10%	65.80%	68.50%
DataStories(Bi-LSTM+Attention)[33]	68.10%	65.10%	67.70%
BERT(Proposed model)	71.44%	71.35%	71.05%

Table 4.4: Results for Twitter Sentiment Analysis

According to SemEval 2017[33], there are two teams who scored the same and where placed first in the rankings for the SemEval 2017 task 4 subtask A. The proposed model has outperformed the other two systems; the average recall of the proposed model is 71.44%, whereas BB-twtr and DataStories both only have 68.10%. The reasons for the improved model performance are provide in the section below.

Discussion

There are two primary reasons for this: the first is a tremendous amount of training material BERT is trained on, and the second is the transformer architecture, as we discussed in previous sections. But perhaps most importantly, the fact that it can capture context information between words while bi-directionality is the most critical feature for BERT while LSTM based models can only recognize sequences in one direction, which can't really capture two direction information at the same time. As we can discussed in the results section both the models BB-twtr and DataStories are based on LSTM.

4.2 Aspect Based Sentiment Analysis (ABSA)

In this section we will be discussing various methods implemented to perform ABSA and how these methods can be used to increase the performance of sentimental analysis models.

4.2.1 Experiment II - Targeted Aspect Based Sentiment Analysis (TABSA) on Sentihood Dataset

We will use TABSA to conduct the experiment. A typical TABSA sentence consists of the following series of words : $\{w_1, \dots, w_n\}$ and some of the words are pre-identified targets $\{t_1, \dots, t_n\}$, following [34], we set the task as 3 class(positive, negative, none) classification problem : given a sentence s, a set of target entities T and a fixed aspect set $A = \{\text{general, price, transit, location, safety}\}$, predict the sentiment polarity $y \in \{\text{positive, negative, neutral}\}$ over the full set of target-aspect pairs. As, we can in Table 4.5 below, the gold standard polarity of (LOCATION2, price) is negative, while the polarity of (LOCATION1, price) is none[36].

Sentihood Dataset Description

TABSA task can be accomplished with the help of the SentiHood dataset. It is derived from the text that was taken from the question-answering platform known as Yahoo! Answers and then filtered for questions that are related to the neighbourhoods that make up the city of London. Location entity names are masked by LOCATION1 and LOCATION2 in the whole dataset, so the task does not involve identification and segmentation of the named entities[34]. An example from Sentihood dataset is given below.

Target	Aspect	Sentiment
LOCATION1	general	positive
LOCATION1	price	none
LOCATION1	price	none
LOCATION1	transit-location	none
LOCATION2	general	none
LOCATION2	price	negative
LOCATION2	price	none
LOCATION2	transit-location	positive

Table 4.5: Example from Sentihood datset

For our experiments we will be converting the ‘none’ to ‘Neutral’ as this is one of the standard class selected in 3 way classification.

Methodology

As discussed in section 4.1.3, we will be using the same BERT configuration. The BERT model selected for this experiment is *BERT-base-uncased*. The hyper-parameters set for this

experiment are as follows :

- Batch size : 64
- Learning rate : 3e-5
- Epochs for training: 2

Majority of the changes for this experiment are done using feature engineering. For this report we will be using BERT's classification task involving sentence pairs, like that of Question Answering (QA) for improving the model's performance. We will construct auxiliary sentences to associate aspects and the target pairs to the reviews. The method proposed for doing this is given below :

BERT - Target (T)ABSA The sentence we want to make from the target aspect pair is a question, and the format will be the same for all the generated sentences. For example, if we consider the target-aspect pair (LOCATION1, dining), the sentence that will be generated is as follows : "what do you think of dining of LOCATION1 ?". The general format for the sentence is as follows : "what do you think of {aspect} of {target} ?"

id	text	auxiliary_sentence	sentiment
0	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the dining of LOCATION1?	2
1	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the general of LOCATION1?	2
2	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the green-nature of LOCATION1?	2
3	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the live of LOCATION1?	2
4	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the multicultural of LOCATION1?	2
5	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the nightlife of LOCATION1?	2
6	1430 LOCATION1 is transforming and the prices will go up and up	what do you think of the price of LOCATION1?	1

Figure 4.3: Example of Sentihood reviews with Auxiliary sentence

Results

We compare our models with the following models :

- LR[34] : A classifier based on logistic regression that takes into account n-grams and pos-tagging.
- LSTM-Final[34] : A biLSTM model with the final state serving as the representation.
- BERT-pair-QA-M[36] : Similar model as our proposed model but the processing done using an older BERT model version.

Performance for the proposed model on Sentihood and its comparison with the other models mentioned above is given in Table 4.6.

Experiment	Accuracy
LR [34]	87.5%
LSTM-Final[34]	82.0%
BERT-pair-QA-M[36]	93.6%
BERT-TABSA(Proposed Model)	96.21%

Table 4.6: Performance on Sentihood dataset

From Table 4.6 it can be observed that our proposed models BERT-TABSA performed better than all the previous models. The proposed model achieved an accuracy of 96.21% which is 2.6% better than the previous best performing model.

Discussion

Why was the proposed model *BERT-TABSA* better than the existing models as in table 4.6. It might be because of the fact that we convert our task to a sentence-pair classification task by converting the target and aspect information into auxiliary sentence which results in significantly increases the size of our original data set. The results, however, demonstrate the significant improvement that BERT has made in its performance when tackling classification tasks.

4.2.2 Experiment III - ABSA on SemEval-2014 Restaurant Dataset

In this section we will be using the SemEval - 2014 Restaurant dataset to perform ABSA.

Dataset Description

Table 4.7 contains the sizes of the training data as well as the test data. The training data for restaurants. Annotations for overall sentence polarities as well as coarse aspect categories were included in this dataset. The training and test data details are provided in Table 4.7. Additional restaurant reviews were gathered, annotated (starting from scratch), and used as test data in the same manner as previously described (800 sentences).

Dataset	Positive	Neutral	Negative	Total
Train	2179	500	839	3550
Test	657	94	159	973

Table 4.7: Polarity distribution SemEval-2014 Restaurant dataset

The below snapshot in Figure 4.4 shows the auxiliary sentences created for the aspects present in the datasets. A review example is also provided in Appendix A.1, here we can see how the annotated data is provided in Extensible Markup Language (XML) format by the SemEval-2014 organizers. An data snippet that corresponds to generated sentence, the review text and the sentiment associated with it is shown below.

ID	sentiment	Aux_Sentence	Review_text
0	3121	2 what do you think of the anecdotes of it ?	But the staff was so horrible to us.
1	3121	2 what do you think of the food of it ?	But the staff was so horrible to us.
2	3121	2 what do you think of the ambience of it ?	But the staff was so horrible to us.
3	3121	1 what do you think of the service of it ?	But the staff was so horrible to us.
4	2777	2 what do you think of the price of it ?	To be completely fair, the only redeeming fact...
5	2777	1 what do you think of the anecdotes of it ?	To be completely fair, the only redeeming fact...
6	2777	0 what do you think of the food of it ?	To be completely fair, the only redeeming fact...

Figure 4.4: Example of Restaurant review with Auxiliary sentence

For our experiment we will only consider the aspect-category for polarity classification as it consists of sentiment for both aspect and the overall document.

Methodology

We use the similar feature engineering as that of Experiment-II but here we will be only considering the aspect. This is equivalent to learning subtask 4 of SemEval-2014 Task 4[32] Aspect-sentiment polarity classification. The sentence generation method is given below.

Restaurant-QA Here, we want to generate from the aspect is a question. We will only consider aspect-category to be associated to the document. For example, the question generated will be “what do you think of the service ?”. All the sentences generated will have the following format “what do you think of {aspect-category} ?”

Results

We compare the results obtained with the two best performing models in SemEval-2014[32].

Experiment	Accuracy
LSTM [32]	82.0%
ATAE-LSTM[32]	84.0%
BERT-pair-QA-M[36]	89.3%
Restaurant-QA(Proposed Model)	92.53%

Table 4.8: Performance on SemEval 2014 Restaurant dataset

Discussion

As we can see from table 4.8 our proposed model *Restaurant-QA* performed better than all the benchmark models. In this experiment we only associate the given aspects to the reviews. Here, sentiment is given to all the aspects which results in our train data to increase in size.

This lets the model know which polarity to assign to the given aspect. This results in the better performance of the model as we have observed from the results.

4.2.3 Experiment IV - ABSA on SemEval-2016 Laptop Dataset

Dataset Description

Download links can be found on the SemEval 2016 task 5 webpage for the datasets that are utilised in the training of the classifiers. This project uses an English dataset in the Laptop domain, which includes the training dataset, testing datasets which is the gold standard. The structure of the dataset is provided in the following table 4.9.

Dataset	Number of Texts	Number of Sentiment	Number of Tuples
Train	450	2500	2909
Test	80	808	801

Table 4.9: SemEval-2016 Laptop dataset structure

The category, which is a pair (Entity#Attribute) consisting of an Entity and an Attribute, is divided into 21 Entity Labels and 9 Attribute labels. The data on training and testing do not follow a uniform distribution with regard to the proportion of each class present in the dataset. We use the pair for classification of sentiment for every Entity and Attribute pair.

Methodology

As discussed in section 4.1.3, we will be using the same BERT configuration. We change some of the configuration for training the BERT model, use a small batch size as compared to other experiments as the data is not sufficient to use a large batch size. The hyper-parameters set for this experiment are as follows :

- Batch size : 8
- Learning rate : 3e-5
- Epochs for training: 2

We employ the same feature engineering that was used in Experiment-II, but in this case, we will consider both Entity and the Attribute to generate the auxiliary sentence. In this experiment we only generate auxiliary sentences for the ENTITY#ATTRIBUTE pair. The sentence generation method is given below

Laptop-QA Here, we are generating question using the Entity#Attribute pair provided for each of the input sentence. For example if Entity#Attribute pair is LAPTOP#Operation_Performance the auxiliary sentence that will be generated is “what do you think of operation_performance of the Laptop ?”.

ID		Review_text	Aux_Sentence	sentiment
0	9.0	GET THIS COMPUTER FOR PORTABILITY AND FAST PRO...	what do you think of the operation_performance...	0
1	10.0	the laptop was really good and it goes really ...	what do you think of the general of LAPTOP?	0
2	11.0	the laptop was really good and it goes really ...	what do you think of the operation_performance...	0
3	12.0	i would really recommend to any person out the...	what do you think of the general of LAPTOP?	0
4	13.0	and its really cheap and you wont regret buyin...	what do you think of the price of LAPTOP?	0

Figure 4.5: Example of Laptop review with Auxiliary sentence

In general the sentences will be generated as follows “what do you think of {Attribute} of the {Entity}?” . Few generated sentences are show in Figure 4.5.

An additional experiment is conducted using the *BERT-large-uncased* model to track and evaluate whether or not a larger BERT model yields better accuracy. The number of encoder layers is the main difference between BERT base and BERT large. The BERT base model has a total of 12 encoder layers stacked on top of one another, whereas the BERT large model has a total of 24 encoder layers stacked on top of one another. There are more parameters (weights) and more “attention heads” to account for as the number of BERT-large layers grows. The BERT base includes 110 million parameters and 12 attention heads (which allow each input token to pay attention to other tokens). On the other hand, BERT large features 16 attention heads and 340 million parameters. For comparison, the total hidden layers on BERT-large are 1024 and BERT-base has 768.

Version Name	Hidden Units	Number of Layers	Number of parameters
BERT-base	768	12-layers	110 million
BERT-large	1024	24-layers	340 million

Table 4.10: Different BERT model version properties.

Results

The results obtained from this experiment is compared with the top two benchmark models from SemEval-2016[30]. We compare our models with the following models :

- IIT-TUDA : During the SemEval-16 competition, IIT-TUDA’s sentence-level sentiment classifier received the highest score possible thanks to its use of the laptop dataset. The SVM algorithm was used for both the aspect classifier and the sentiment classifier when applying this method.
- ENCU : On the laptop dataset provided by SemEval-2016 for use in the evaluation of text-level sentiment classifiers, the ENCU model came in first place. They used characteristics from linguistics, a sentiment lexicon, a subject model, and word2vec in order to make a prediction about the sentiment associated with an element. Features and a logistic regression classifier were used to detect text sentiment.

The table 4.11 shows our proposed model performance compared to the two models mentioned above.

Experiment	Accuracy
ECNU[30]	78.152%
IIT-TUDA [30]	82.772%
Laptop-QA(BERT-base model)	82.54%
Laptop-QA(BERT-large model)	84.64%

Table 4.11: Performance on SemEval 2016 Laptop dataset

Discussion

We have used the same type of feature engineering in this experiment the difference here is we have used the ENTITY#ATTRIBUTE pair to generate the auxiliary sentences. As we see from the results obtained our model performance was on par with the benchmark system provided in SemEval-2016. *Laptop-QA(BERT-large model)* performed the best as it got the highest accuracy 1.87% better than the best performing model in the result table 4.11. As BERT-large understands the semantic relationship between words, when we provide the models with extra information about the polarity of the aspect and which attribute it is associated to in form of auxiliary sentences. As BERT-large is a larger model as compared to BERT-base as we discussed in Methodology section, it is able to learn more parameters than BERT-base, which is the result of having more attention heads in BERT-large. The polarity classification has more data to train and is able to improve its prediction on unseen data. The main difference between this experiment and experiment II and III is that we have only considered the aspects that are provided in the dataset not the whole range of aspects are present in the dataset. This is why, we can directly compare the accuracy with that of other model presented in SemEval 2016 conference.

Chapter 5

Conclusion and Future Work

This chapter provides a concise overview of the study by discussing the study's goals and revealing the study's conclusion. Finally, we discuss the results and the implications of this study, as well as the next steps researchers can take to build upon this work.

5.0.1 Conclusion

Sentiment analysis has tremendous value when applied to practical commercial applications. The research interest in this is also very high as compared to other NLP research areas. For example, in an e-commerce context, sentiment analysis is applied to provide product reviews as input for data mining and market research companies like Amazon. Apart from this, sentimental analysis can be used for various other purposes.

For this project, we focused on sentimental analysis on two different levels. The first is Twitter sentiment analysis which is a document-level analysis, and the second is aspect-based sentiment analysis. In every chapter, we divide it into two parts, dedicating the first to Twitter sentiment analysis and the second to aspect-based sentiment analysis. The goal of this study was to use document level sentiment analysis and Aspect-Based Sentiment Analysis to improve upon the current benchmark for pre-trained models that make use of the Transformer architecture. The main issue with performing Twitter sentiment analysis was that the data that was provided contained generalized data, i.e., it was not domain specific. Existing models require a large amount of data to train to give a good performance on these types of datasets, but in our case, as we utilized BERT, which is a pre-trained model, these models are trained on a tremendous amount of data, this gives these models a huge advantage when it comes to NLP tasks. We need to fine tune the model to our specific tasks. That's what gave our proposed model a big performance jump as compared to other benchmark models from SemEval-2017 Task 4. We got a performance increase in accuracy of 6.25% than the best model given in SemEval-2017 Task 4.

The second part focused on Aspect-based sentiment analysis (ABSA). For ABSA, we mainly focused on feature engineering tasks and experiments. We structured the data such that our proposed model is able to extract maximum information from a limited domain

specific dataset. The main approach used for ABSA was to construct auxiliary sentences and provide details about the aspect in form of a question. We used a similar approach on various datasets to verify our method. It performed reasonably for all the SemEval datasets, providing state-of-the-art results for all of the experiments performed. As we have seen in section 4.2. Our proposed model achieved 3.23% better accuracy on SemEval-2014 Restaurant dataset. The same can be said about the SemEval-2016 Laptop Dataset, where our model achieved nearly 2% better performance than the previous model that was regarded to be the best performing in SemEval-2016 conference.

5.0.2 Future Work

The method of generating auxiliary sentences to associate aspects to their target/entity shows us the validity of the method. We can use the same approach on various other ABSA datasets to perform similar tasks. For future work, we can also focus on aspect-extraction and aspect-polarity-detection tasks which are introduced in SemEval-2014 Task 4[32]. This tasks if implemented correctly this can let researchers use the datasets which do not have the aspects and polarity mapped to the reviews. These types of datasets are widely available and have large amount of data points. In addition to this, numerous researchers have developed their own unique iterations of the BERT model, such as RoBERTa, BART, DistilBERT, DeBERT. Each of these models possesses a unique set of characteristics and operates in a manner that is distinct from the others. The proposed feature engineering and the idea that was proposed in this project can be used to apply to those various pre-trained models. After that, we will be able to compare the performance of these models on similar tasks. In terms of the model improvement, in this report we have used a feed-forward neural network with a single hidden layer for sentiment classification. We can replace this with a CNN layer to observe if we get any improvement in model performance.

Bibliography

- [1] Huggingface - Transformer library. <https://huggingface.co/docs/transformers/index>. Accessed: 2022-08-17.
- [2] Pytorch. <https://pytorch.org/>. Accessed: 2022-08-17.
- [3] Scikit-learn. <https://scikit-learn.org/>. Accessed: 2022-08-17.
- [4] Semeval official website. <https://semeval.github.io/>. Accessed: 2022-08-14.
- [5] LSTM architecture and explanation. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. Accessed: 2022-08-16.
- [6] NLTK. <https://www.nltk.org/>. Accessed: 2022-08-17.
- [7] COHEN, V., AND GOKASLAN, A. OpenGPT-2: Open language models and implications of generated text. *XRDS* 27, 1 (sep 2020), 26–30.
- [8] COLLOBERT, R., WESTON, J., BOTTOU, L., KARLEN, M., KAVUKCUOGLU, K., AND KUKSA, P. Natural language processing (almost) from scratch. *Journal of machine learning research* 12, ARTICLE (2011), 2493–2537.
- [9] DEVLIN, J., CHANG, M., LEE, K., AND TOUTANOVA, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).
- [10] DOS SANTOS, C., AND GATTI, M. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (Dublin, Ireland, Aug. 2014), Dublin City University and Association for Computational Linguistics, pp. 69–78.
- [11] ELMAN, J. L. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.
- [12] GIACHANOU, A., AND CRESTANI, F. Like it or not: A survey of twitter sentiment analysis methods. *ACM Comput. Surv.* 49, 2 (jun 2016).
- [13] HAN, X., ZHANG, Z., DING, N., GU, Y., LIU, X., HUO, Y., QIU, J., YAO, Y., ZHANG, A., ZHANG, L., ET AL. Pre-trained models: Past, present and future. *AI Open* 2 (2021), 225–250.

- [14] HOANG, M., BIHORAC, O. A., AND ROUCES, J. Aspect-based sentiment analysis using BERT. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics* (Turku, Finland, Sept.–Oct. 2019), Linköping University Electronic Press, pp. 187–196.
- [15] HOCHREITER, S., AND SCHMIDHUBER, J. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [16] HU, M., AND LIU, B. Mining and summarizing customer reviews. 168–177.
- [17] KOLKUR, S., DANTAL, G., AND MAHE, R. Study of different levels for sentiment analysis.
- [18] LAI, S., XU, L., LIU, K., AND ZHAO, J. Recurrent convolutional neural networks for text classification. In *Twenty-ninth AAAI conference on artificial intelligence* (2015).
- [19] LECUN, Y., BOTTOU, L., BENGIO, Y., AND HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
- [20] LIU, B. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.
- [21] LIU, B., AND ZHANG, L. A survey of opinion mining and sentiment analysis. *Springer* (2012), 415–463.
- [22] MURTY, M., AND DEVI, V. *Pattern recognition. An algorithmic approach.* 01 2011.
- [23] NAKOV, P., KOZAREVA, Z., RITTER, A., ROSENTHAL, S., STOYANOV, V., AND WILSON, T. Semeval-2013 task 2: Sentiment analysis in twitter. *CoRR abs/1912.06806* (2019).
- [24] NAKOV, P., RITTER, A., ROSENTHAL, S., SEBASTIANI, F., AND STOYANOV, V. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 1–18.
- [25] NGUYEN, T. H., AND SHIRAI, K. Topic modeling based sentiment analysis on social media for stock market prediction. 1354–1364.
- [26] O’SHEA, K., AND NASH, R. An introduction to convolutional neural networks. *CoRR abs/1511.08458* (2015).
- [27] PAK, A., AND PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining.
- [28] PANG, B., LEE, L., ET AL. Opinion mining and sentiment analysis. *Now Publishers, Inc.* (2008), 1–135.

- [29] PETERS, M. E., NEUMANN, M., IYYER, M., GARDNER, M., CLARK, C., LEE, K., AND ZETTLEMOYER, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (New Orleans, Louisiana, June 2018), Association for Computational Linguistics, pp. 2227–2237.
- [30] PONTIKI, M., GALANIS, D., PAPAGEORGIOU, H., ANDROUTSOPoulos, I., MANANDHAR, S., AL-SMADI, M., AL-AYYOUB, M., ZHAO, Y., QIN, B., DE CLERCQ, O., HOSTE, V., APIDIANAKI, M., TANNIER, X., LOUKACHEVITCH, N., KOTELNIKOV, E., BEL, N., JIMÉNEZ-ZAFRA, S. M., AND ERYİĞIT, G. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (San Diego, California, June 2016), Association for Computational Linguistics, pp. 19–30.
- [31] PONTIKI, M., GALANIS, D., PAPAGEORGIOU, H., MANANDHAR, S., AND ANDROUTSOPoulos, I. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (Denver, Colorado, June 2015), Association for Computational Linguistics, pp. 486–495.
- [32] PONTIKI, M., GALANIS, D., PAVLOPOULOS, J., PAPAGEORGIOU, H., ANDROUTSOPoulos, I., AND MANANDHAR, S. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (Dublin, Ireland, Aug. 2014), Association for Computational Linguistics, pp. 27–35.
- [33] ROSENTHAL, S., FARRA, N., AND NAKOV, P. SemEval-2017 task 4: Sentiment analysis in Twitter. 502–518.
- [34] SAEIDI, M., BOUCHARD, G., LIAKATA, M., AND RIEDEL, S. Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. *arXiv preprint arXiv:1610.03771* (2016), 1546–1556.
- [35] SASSANO, M. Virtual examples for text classification with support vector machines. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (2003), pp. 208–215.
- [36] SUN, C., HUANG, L., AND QIU, X. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *CoRR abs/1903.09588* (2019).
- [37] THAKKAR, H., AND PATEL, D. R. Approaches for sentiment analysis on twitter: A state-of-art study. *CoRR abs/1512.01043* (2015).
- [38] TURNEY, P. D. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *CoRR cs.LG/0212032* (2002).

- [39] VASWANI, A., SHAZER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L., AND POLOSUKHIN, I. Attention is all you need. *Advances in neural information processing systems 30* (2017).
- [40] WANG, B., LIAKATA, M., ZUBIAGA, A., AND PROCTER, R. TDParse: Multi-target-specific sentiment recognition on Twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers* (Valencia, Spain, Apr. 2017), Association for Computational Linguistics, pp. 483–493.
- [41] WU, Y., SCHUSTER, M., CHEN, Z., LE, Q. V., NOROUZI, M., MACHEREY, W., KRIKUN, M., CAO, Y., GAO, Q., MACHEREY, K., ET AL. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [42] ZHANG, H. The optimality of naive bayes. *Aa 1, 2* (2004), 3.

Appendices

Appendix A

Supporting Images for the report

A.1 BERT Sentiment classification architecture

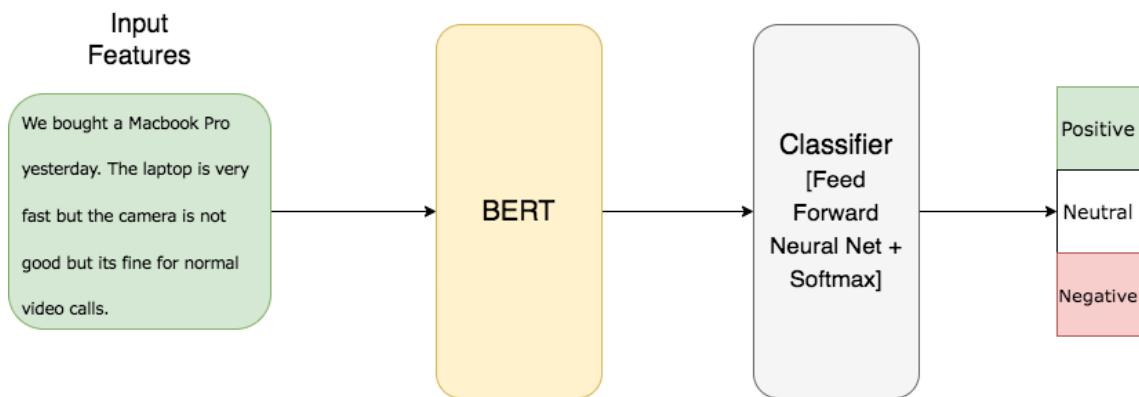


Figure A.1: BERT Pipeline for Sentiment Analysis

A.2 XML Snippet for SemEval-2014 Restaurant Dataset

```
<sentence id="3121">
  <text>But the staff was so horrible to us.</text>
  <aspectTerms>
    <aspectTerm term="staff" polarity="negative" from="8" to="13"/>
  </aspectTerms>
  <aspectCategories>
    <aspectCategory category="service" polarity="negative"/>
  </aspectCategories>
</sentence>
```

Figure A.2: XML snippet for SemEval-2014 Restaurant Dataset

Acronyms

ABSA Aspect-based sentiment analysis

BERT Bidirectional Encoder Representations from Transformers

NLP Natural Language Processing

SA Sentiment Analysis

XML Extensible Markup Language