

Sentiment Analysis Using Pretrained Bidirectional Encoder Representation from Transformer

Sahar Moarefi

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
February 2024
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Zeki Bayram
Chair, Department of Computer
Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Adnan Acan
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Adnan Acan

2. Assoc. Prof. Dr. Mehtap Köse Ulukök

3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

Sentiment analysis is a quickly expanding area of research in natural language processing and text classifications. This technique has become an essential part of a wide range of applications such as politics, business, advertising and marketing. In order to achieve stronger marketing, checking and analyzing users' opinions in the digital world has become a basic need. Therefore, the use of more up-to-date techniques with a higher percentage of accuracy will create targeted marketing and success in sales and businesses in the real world. There are various approaches for sentiment analysis, but in this thesis the most popular and effective model by focusing on Sentiment Analysis study which is RoBERTa model to detect emotions in extensive textual data is discussed.

Experimental results demonstrate the RoBERTa model's application on three datasets—IMDB (50,000 movie reviews) and Sentiment140 and Disneyland Reviews . Due to hardware constraints, pre-trained models mitigate the need for extensive data and computational resources. Evaluation metrics, including accuracy, precision, recall, and F1-score, attest to the model's effectiveness that has represented in chapter 5.

The thesis contributes insights into the evolution of sentiment analysis, underscores the significance of transformer-based models, and provides empirical evidence of RoBERTa's efficacy in sentiment analysis across diverse datasets. It emphasizes the pivotal role of advanced NLP techniques in achieving accurate and efficient sentiment analysis in contemporary applications.

Keywords: NLP, sentiment analysis, transformers, RoBERTa

ÖZ

Duygu analizi, doğal dil işleme ve metin sınıflandırmalarında hızla genişleyen bir araştırma alanıdır. Bu teknik siyaset, iş dünyası, reklamcılık ve pazarlama gibi çok çeşitli uygulamaların vazgeçilmez bir parçası haline geldi. Daha güçlü bir pazarlamaya ulaşmak için dijital dünyada kullanıcıların görüşlerinin kontrol edilmesi ve analiz edilmesi temel bir ihtiyaç haline geldi. Bu nedenle, daha güncel tekniklerin daha yüksek doğruluk yüzdesiyle kullanılması, gerçek dünyada satışlarda ve işletmelerde hedefe yönelik pazarlama ve başarı yaratacaktır. Duygu analizi için çeşitli yaklaşımlar vardır ancak bu tezde, Duygu Analizi çalışmasına odaklanılarak en popüler ve etkili model olan, kapsamlı metinsel verilerdeki duyguları tespit etmeye yönelik RoBERTa modeli tartışılmıştır.

Deneyisel sonuçlar, RoBERTa modelinin üç veri kümesine (IMDB 50.000 film incelemesi) ve Sentiment140 ve Disneyland İncelemeleri) uygulanmasını göstermektedir. Donanım kısıtlamaları nedeniyle önceden eğitilmiş modeller, kapsamlı veri ve hesaplama kaynaklarına olan ihtiyacı azaltır. Doğruluk, kesinlik, geri çağırma ve F1 puanı dahil değerlendirme ölçümleri, modelin 5. Bölümde sunulan etkinliğini kanıtlamaktadır.

Tez, duyarlılık analizinin evrimine dair içgörülere katkıda bulunuyor, transformatör tabanlı modellerin öneminin altını çiziyor ve RoBERTa'nın çeşitli veri kümeleri genelinde duyarlılık analizindeki etkinliğine dair ampirik kanıtlar sağlıyor. Çağdaş uygulamalarda doğru ve etkili duygu analizi elde etmede ileri NLP tekniklerinin önemli rolünü vurgulamaktadır.

Anahtar Kelimeler: doğal dil işleme, duygu analizi, transformatörler, RoBERTa

DEDICATION

To my beloved husband, your unwavering support has been my guiding light throughout this academic journey.

ACKNOWLEDGMENT

I would like to express my deepest appreciation to my supervisor, Assoc. Prof. Dr. Adnan Acan, for his exceptional guidance, unwavering support, and invaluable insights throughout the completion of this thesis. I am grateful for the time, dedication, and constructive feedback provided, which significantly enriched the quality of this work. I am fortunate to have had such a dedicated and inspiring supervisor, and I sincerely thank him for his mentorship.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	iv
DEDICATION	vi
ACKNOWLEDGMENT	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
1 INTRODUCTION	1
1.1 Natural Language Processing	1
1.2 Sentiment Analysis	2
2 LITERATURE REVIEW	4
2.1 The Advancement of Methods for Sentiment Analysis	4
3 METODOLOGY	7
3.1 Convolutional Neural Network	7
3.1.1 CNN Architecture	8
3.2 Recurrent Neural Network	10
3.2.1 RNN Architecture	11
3.3 Long Short-Term Memory	12
3.3.1 LSTM Architecture	13
4 ALGORITHMIC AND ARCHITECTURAL DESCRIPTIONS	15
4.1 Transformers	15
4.2 Bert Architecture.....	17
4.3 RoBERTa Architecture	21
5 EXPERIMENTAL RESULTS	26

5.1 Selected Model for Experimental Results.....	26
5.2 Datasets	27
5.2.1 IMDB Dataset of 50k Movie Reviews	27
5.2.2 Sentiment140 Dataset with 1.6 Million Tweets	27
5.2.3 Disneyland Reviews.....	28
5.3 Experimental Setup	28
5.3.1 Software Environment	28
5.3.2 Hardware Configuration	28
5.4 The Metrics Used for Evaluation	29
5.5 Algorithmic Steps for Sentiment Analysis by RoBERTa	29
5.6 Results	30
5.6.1 IMDB Dataset Results	30
5.6.2 Twitter Dataset Results	31
5.6.3 Disneyland Reviews Dataset Results	33
5.6.4 Discussion	34
5.6.5 Comparison the Same Datasets with Different Models.....	36
6 CONCLUSION	38
REFERENCES	40

LIST OF TABLES

Table 1: Test Results of All Used Datasets	34
Table 2: Comparison Other Models with IMDB Dataset.....	36
Table 3: Comparison Other Models with Sentiment140 Dataset	37
Table 4: Comparison Other Models with Disneyland Reviews Dataset	37

LIST OF FIGURES

Figure 1: Sentiment Analysis Process on Product Reviews	3
Figure 2: CNN Based Text Classification Model	6
Figure 3: Approach of Sentiment Analysis	6
Figure 4: CNN's Structure	10
Figure 5: Comparison Between RoBERTa Base and Large.....	27
Figure 6: Representation Of Training Accuracy on IMDB	30
Figure 7: Representation of Training Loss Function on IMDB	30
Figure 8: Confusion Matrix on IMDB	31
Figure.9 Representation of Training Accuracy on Twitter	31
Figure 10: Representation of Training Loss Function on Twitter	32
Figure 11: Confusion Matrix on Twitter	32
Figure 12: Representation of Training Accuracy on Disneyland Reviews.....	33
Figure 13: Representation of Training loss Function on Disneyland Reviews.....	33
Figure 14: Confusion Matrix on Disneyland Reviews	34

Chapter 1

INTRODUCTION

1.1 Natural Language Processing

Natural language is called the language of us, humans, to communicate with each other, but in contrast communication between computers is not as easy as human communication to understand and interpret, so this is where NLP which is an interdisciplinary science between artificial intelligence and linguistics is entered. Due to the significant progress of science, natural human languages can be processed using artificial intelligence techniques and used in various applications. Natural language Processing holds enormous importance due to its revolutionary affection on various aspects of human-computer interaction.

Natural language processing is a wide area with various techniques and approaches that includes everything from understanding a sentence's grammatical structure to figuring out a paragraph's underlying sentiment. These techniques can be categorized into several subfields such as Syntax and Grammar Analysis, Machine Translation, Question Answering, Speech Recognition, Sentiment Analysis and etc.

Computational linguistics, deep learning models, and machine learning techniques are all integrated in Natural Language Processing. NLP breaks down the linguistic information into smaller units called "Tokens" so that each word in the phrase is treated as a separate unit, aided by massive datasets that are pre-obtained. Tokenization is the

technique that forms the basis of NLP technology. In order to comprehend the meaning of the text and the intentions behind it, the computer then looks for a relationship between these tokens. Ongoing research and innovation in Natural Language Processing promise a future where machines not only understand the complexities of language, but also adapt and learn from different linguistic contexts.

Consequently, the journey of Natural Language Processing is to bridge the communication gap between man and machine. With the pervasiveness of transformer architecture and transfer learning patterns, the NLP landscape is poised for further advancements in various areas such as sentiment analysis, which will be discussed further.

1.2 Sentiment Analysis

In this era, large amounts of text data are generated daily and understanding the expressed emotions and opinions in the data has become a controversial issue. As the digital age pushes us toward enormous amounts of textual data, the ability to understand and interpret sentiment across this vast expanse has emerged as a critical tool.

Sentiment analysis is known as opinion mining or sentiment artificial intelligence. It refers to affective states and subjective information using natural language processing, text analysis, computational linguistics, and biometrics. The purpose of sentiment analysis is to determine writer attitudes toward a special topic or product. Sentiment analysis provides companies with a means to estimate the extent of product acceptance and to determine strategies to improve product quality [1]. Therefore, machines that

are enabled to predict context sentiment on different platforms can be used to achieve positive, negative, and neutral attitudes.

Sentiment Analysis is a natural language processing approach which can be considered a classification process that the main goal of it is to determine the emotion whether the written text is positive or negative as illustrated in Figure 1 [2]. Analyzing the reviews and looking at the sentiment scores is the major purpose of sentiment analysis.

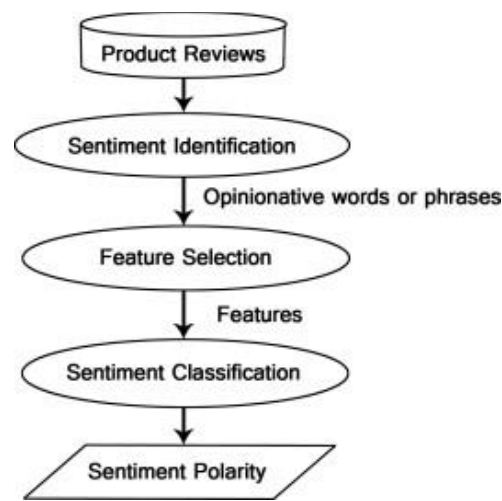


Figure 1: Sentiment analysis process on product reviews

This thesis provides a general overview of the approaches which have been used and the ones that are more popular recently of Sentiment Analysis to overcome the challenges in the past. Besides, in this thesis just two positive and negative attitudes are used by the RoBERTa model [3]. The technique which has used in Sentiment Analysis offer insights into the efforts made to gain accurate and efficient results.

Chapter 2

LITERATURE REVIEW

2.1 The Advancement of Methods for Sentiment Analysis

The technology used for language processing in the 1990s was very different from today's advanced approaches. The easiest way to collect the positive or negative words was to count the words with their positive and negative meanings in the sentences. That solution was simple but it had problems such as there were words with a negative meaning in a sentence with a positive meaning, which made it difficult or erroneous to recognize whether the sentence was positive or negative.

In an attempt to address the problem, engineers created complex handcrafted rules in the early 2000s and attempted to codify English language grammar. Over the next decade, machine learning replaced manual coding .The basic algorithms used in machine learning sentiment analysis are a combination of manual codes and weights.

In the middle of the 2010s, deep learning techniques, including neural networks, were introduced in the sentiment analysis space. Before deep learning became widely utilized, sentiment analysis used traditional machine learning techniques such as Support Vector Machines (SVM) [4],[5], Naive Bayes [6], and rule-based methods that because of their limited ability to capture complex patterns in natural language they used handcrafted features.

In 2013, a group of Google researchers under the direction of Tomáš Mikolov created word vector embedding methods such as Word2vec and GloVe which they were become a standard for some problems of the NLP, incorporating sentiment analysis and classification of texts [7]. In the lexicon for every word, a vector is added in Cartesian space as part of the word embedding process. In vector space, every word with a similar meaning has a close vector to each other, although the vectors of words with different meanings are completely separate from each other. By capturing the semantic relations between words, these embeddings helped models learn greater understanding of language representations.

By utilizing word embedding approaches, sentences can be converted to a matrix of numbers and then passed as input into a convolutional neural network [8],[9] or another deep neural network architecture for instance, Recurrent neural network [10] or Long Short-Term Memory networks [11].

Word embedding is one method for converting text words into digital vectors to text classification in the CNN's networks. To analyze them using the conventional machine learning algorithm, these vectors must be transformed into numbers and provided as digital input. To store them a $V \times D$ matrix is needed. D stands for the word embedding dimension, and V stands for the vocabulary size. The matrix is called the embedding layer. The model can be constructed using a combination of pooling and convolutional layers, depending on the classification performance indicators and text length. In general, figure.2 illustrates the CNN text classification model's structure.

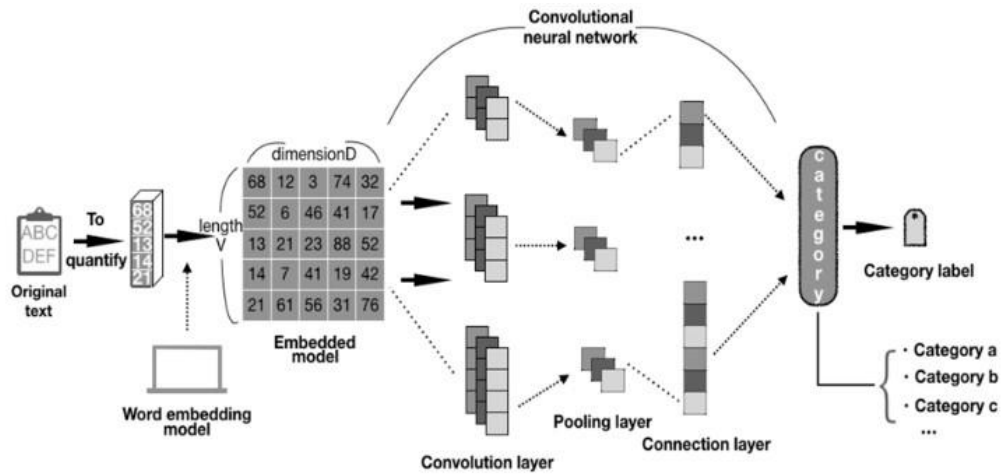


Figure 2: CNN-Based Text Classification Model [9]

The current approach is transformer-based which provides promising results in sentiment analysis. Transformers convert words into vectors in a massive multiple-dimensional area, much like word vector embeddings do [12] Figure 3. The most popular transformers which are used in this field are BERT [13] and RoBERTa [14].

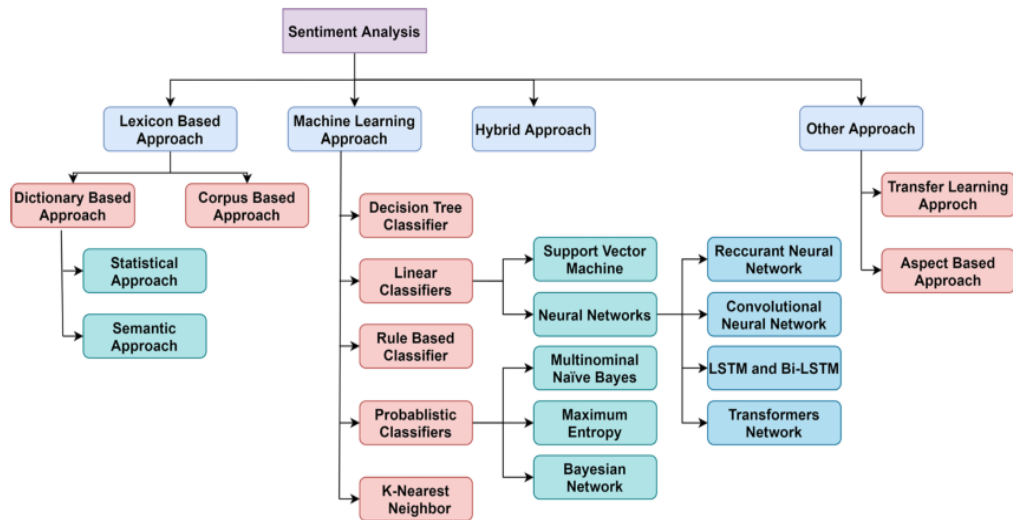


Figure 3: Approach of Sentiment Analysis

Chapter 3

METHODOLOGY

3.1 Convolutional Neural Network

CNN stands for Convolutional Neural Network as an effective type of artificial neural network that designed for processing structured data, such as images or sound related to computer vision. Although it is originally designed for computer vision tasks, it has found applications in Natural Language Processing. The most important usage of CNN in NLP is for text classification and sentiment analysis. The fundamental feature of CNN is the ability to learn data directly from raw input automatically and adaptively hierarchical representations. In the same way that convolutional operations can identify visual patterns in images, they can also identify local patterns and features in the input text. Convolutional layers, which apply convolution operations to input data, are used to achieve this.

While CNNs have been successful in Natural Language Processing tasks, their popularity has decreased in compare to other architectures like RNNs and transformer models. One reason for this reduction is the lack of sequential modeling because it is better designed to capture local patterns rather than sequential modeling. Moreover, CNN is not as strong as other neural networks due to more requirement understanding the contextual relationships between words in a sentence; also, is has struggle to capture long-term dependencies in sequences which are crucial for language modeling

and understanding the context of a text.[15] Other mechanisms may perform better in some cases, but CNN has well functionality in tasks such as text classification.

3.1.1 CNN architecture

Convolutional Neural Networks have traditionally been associated with computer vision tasks, but their application to text data has proven effective in tasks such as sentiment analysis and text classification. A detailed overview of CNN's structure is available in the Figure 4, natural language processing scenario are not fully covered [16].

The architecture of a CNN for text data involves the following components:

- Input Layer;

Represents the input text, where each word is embedded into a high-dimensional vector.

- Convolutional Layers;

Convolutional layers are fundamental components of Convolutional Neural Networks (CNNs) designed to extract features from input data, thus they are applied to embedded word vectors.

In the context of text processing, these layers employ filters, also known as kernels, that convolve across the input to detect patterns and capture relevant features. For text data, they identify important n-grams or sequential patterns. To find local patterns, filters with certain weights move over the input sequence. By capturing distinct information simultaneously through several filters, the model is able to produce hierarchical representations.

- Pooling Layers;

Max pooling or average pooling reduces the dimensionality of the feature maps. Pooling focuses on the most salient features, enhancing computational efficiency and reducing overfitting.

- Activation Function:

Activation functions, such as Rectified Linear Unit, are commonly applied to introduce non-linearity that it helps capture complex relationships also allows the model to learn from diverse patterns.

- Loss Function:

The model's prediction error during training is measured by the loss function. Reducing the loss function is the aim of CNN training, which helps the network generate accurate predictions.

- Optimization:

The goal of optimization methods is to identify the ideal weight configuration that enables the CNN to produce accurate predictions based on the provided text data.

- Fast Processing:

It refers to the efficiency with which the model can analyze and make predictions on large volumes of text data, especially in real-time applications, to ensure quick and timely responses.

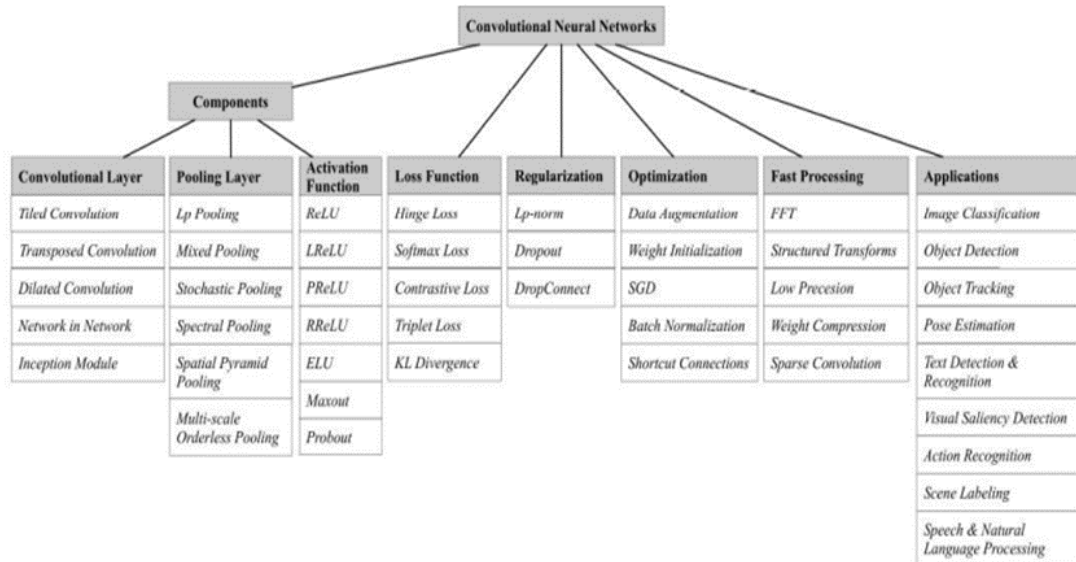


Figure 4: CNN's Structure [16]

Convolutional layers enable CNNs to capture hierarchical characteristics and local patterns in text input, that makes them useful for sentiment analysis.

3.2 Recurrent Neural Network

Another class of neural network's architectures, recurrent neural networks (RNNs), has gained more attention from researchers for modeling sequences and have been shown to perform well in solving various NLP issues due to their ability to deal with capture dependencies and patterns over time. The important feature of RNNs is their usage of recurrent connections structure, that allows information to be passed along from one step of the sequence to the next one. This recurrent structure makes it possible to maintain the hidden state and encapsulate the information about the previous entries in the sequence. The hidden state is a memory that records the context and dependencies in different time steps.

Nevertheless, RNNs have certain drawbacks, such as being prone to experience vanishing or exploding gradient issues during training and difficulty capturing long-

range dependencies. When the neural network is trained, first the inputs are given to the model, then the output is produced. This output is compared with the actual value and the error is obtained, then the weights are updated. If the change of weights is small, the last weights have been changed in the return; But the weights of the initial layers of the network do not change. This issue causes the initial weights of the network not to be updated or changed in very small errors, and this happens in the term Vanishing. The concept of Exploding happens when the weights of the last layers are too big and when they are updated, they are multiplied by the weights of other layers. As a result, very large initial weights are produced. In addition, it discards the information of the previous steps and has close memory. The RNN architectures, such as Long Short-Term Memory networks (LSTMs), have been developed to address these problems, although RNN is still employed in certain applications due to its understanding in the sequential data processing.

3.2.1 RNN architecture

RNN is designed to process sequential data for converting an input pattern into an output pattern. The structure enables the networks to capture the temporal dynamics and perform sequential prediction. There are hidden states works as the memory that captures information from previous steps in a sequence.

- Sequential Processing

RNNs sequentially process each element of the incoming data one at a time. These elements can be letters, words, or word embeddings in the context of text data.

- Hidden State

RNNs keep track of a hidden state that serves as the network's memory, storing details about earlier components in the series. Based on the earlier hidden state and the current input, the hidden state is updated at each step.

- Parameter Sharing

The sharing of parameters between various time steps is a crucial characteristic of RNNs. Since the same set of weights is applied at each step, the network is able to recognize and identify Table 1. ResultsDependencies and patterns in sequential data are intended to be learned by RNNs. They work effectively for situations where it's important to comprehend context from recent items since they can capture short-term dependencies well.

RNNs successively process text data while maintaining a hidden state to extract information from earlier stages. Advanced RNN architectures are frequently chosen for applications requiring the understanding of long-term relationships in text sequences, even though they are useful for short-term dependencies as well.

3.3 Long Short-Term Memory

In 1997, Sepp Hochreiter and Jürgen Schmidhuber introduced LSTM. Long Short-Term Memory, or LSTM for short, is a kind of recurrent neural network architecture that was created to get within some of the issues that traditional RNNs had when it came to capturing long-term dependencies in sequential data. The main challenge has solved by LSTM is the vanishing gradient problem, which is common in traditional Recurrent Neural Networks.

A more complex structure has introduced in this architecture which is cell state. Abstracts of very important information are kept in it. It leads them to selectively remember and forget information over long sequences. Transformers had to be used as a result of these flaws.

3.3.1 LSTM Architecture

Traditional RNNs have limitations, like the vanishing gradient problem, which makes it more difficult for the network to identify long-term dependencies. More advanced RNN architectures, such as Long Short-Term Memory networks, have been designed to overcome this. This architecture incorporate method to choose keep or forget information in the hidden state, allowing them to capture long-range dependencies more efficiently.

To address the difficulties associated with capturing long-range relationships in sequential input and disappearing gradients, the LSTM is made with the following essential parts:

- Cell State (C_t)

LSTM networks create a cell state that maintains through the sequence, enabling the network to add or remove data on demand. It serves as a memory device, keeping track of essential data throughout the sequence.

- Hidden State (H_t)

The hidden state contains data from earlier time steps and resembles the output. It acts as a storage space for the sequence's earlier steps.

- Gates

For controlling the information flow, LSTMs employ Forget, Input and Output gates.

- Cell State Update

To update the cell state, the input gate, forget gate, and prior cell state are combined.

Due to its distinct architecture, the LSTM is ideally suited for applications like emotion analysis, speech recognition, and language modeling because it can efficiently capture dependencies in sequential data over large distances.

Chapter 4

ALGORITHMIC AND ARCHITECTURAL DESCRIPTIONS

4.1 Transformers

A kind of neural network design known as the Transformer was first shown in the 2017 publication "Attention is All You Need," which was written by Vaswani et al [17]. The Transformer architecture is now the foundation for many language-related tasks and has had a significant influence on the field of natural language processing. In many different natural language processing tasks, such as question-answering, text summarization, machine translation, and language modeling, Transformers have shown state-of-the-art performance. Transformer-based models that are noteworthy are T5 as Text-to-Text Transfer Transformer, Generative Pre-trained Transformer, and Bidirectional Encoder Representations from Transformers. [18]

Deep learning approaches have revolutionized the field of data analysis and bioinformatics, laying the groundwork for groundbreaking discoveries and fresh perspectives. Transformer-based architectures and attention mechanisms have shown improved performance and capacities in handling the intrinsic complexity of genome data in recent times, thanks to their development and use. In many fields, including computer vision and natural language processing, deep learning techniques especially those that make use of transformer topologies and attention mechanisms have demonstrated impressive efficacy [19].

There are a number of reasons why transformers became popular in the field of natural language processing. The ability to effectively capture long-range dependencies in sequential data and possibility of parallel processing to overcome the LSTM problem is the most important reasons. This problem has been solved in the transformer and we can give all the words to the transformer at once so that the model can be trained in parallel. With the possibility of parallel processing, all the words can be given to the transformer at once to train the model in parallel. The high parallelizability of transformers means that they can process input sequences in parallel rather than sequentially. This makes transformers more efficient for large datasets and powerful computing resources because they take less time to train and can be scaled up. The parallelization is in contrast to the sequential nature of RNNs due to computations depend on the output of the previous time step.

Transformers can assess the relative importance of various input sequence elements differently because they make use of a self-attention mechanism. The model can effectively capture long-range dependencies and concentrate on pertinent data. Transformers are able to solve the vanishing gradient problem, which is a difficulty for traditional recurrent neural networks, with the aid of the attention mechanism. Transformers convert words in a sentence into vectors in a massive multidimensional space, much like word vector embeddings do. One distinction is that, based on how a word functions in a phrase, a transformer model may represent it as a different vector.

The transformer consists of encoder and decoder parts. While the decoder creates the output sequence, the encoder processes the input sequence. Every stack is made up of several levels, including a position-wise feedforward network and a multi-head self-attention mechanism in each layer.

In Encoder part, random embeddings are given to a multi head attention along with the words' position. Normalization, feed forward and normalization are done again as the next step. After these steps that are repeated N times, the output is entered into the Decoder. Creating the attention vector is the goal of multi-head attention. it indicates what words from the past should be considered to produce the output words. This focus belongs to a sentence. Finding the weights associated with each word's level of importance is what attention is meant to represent.

Currently, the transformer approach represents the state of the art in sentiment analysis. BERT is the most well-known transformer model which is expressed in the following.

4.2 BERT Architecture

Transformers were first introduced by Google in 2017. At that time, language models mainly used Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) to perform NLP tasks. The robust natural language processing approach known as BERT, or Bidirectional Encoder Representations from Transformers, was created by Google. It was first presented by Jacob Devlin and colleagues in a ground-breaking research paper titled "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" in 2018 [20].

Text is processed sequentially by traditional language models, either from left to right or from right to left. The model's awareness is restricted by this strategy to the immediate context that comes before the target word. Instead of examining the text piece by piece, BERT employs a bi-directional method that takes into account both the left and right context of words in a sentence. This allows BERT to examine every word in a sentence at once. Although these models are powerful, Transformer provides a

significant improvement in this area because it does not need to process data sequences in a fixed order, whereas RNN and CNN do. Language models have historically only been able to read text input sequentially, that is, from left to right or from right to left, but not simultaneously. So, BERT is designed to read simultaneously in both directions [21]. Because transformers process data in any order, they enable training on larger amounts of data than before. This possibility led to the creation of pre-trained models such as BERT. Models that were trained on huge amounts of linguistic data. An enormous corpus of unlabeled text data serves as BERT's pre-training dataset. The model learns to anticipate words that are disguised or missing from a sentence during pre-training. BERT is able to recognize general language patterns and contextual correlations because of its unsupervised pre-training step.

A new machine learning paradigm called "transfer learning" focuses on applying knowledge from one task to other tasks that are similar. The pertained BERT model is a new sort of transformer that Google suggested in 2018. Since the BERT language model is sequential. BERT generates contextualized vector representations for each element of the input sequence given an input sequence in a language format. With the help of an encoder, this highly generalizable language representation architecture completes its mission. Text representations that are encoded are produced using encoders, a neural network design that is derived from the transformer. There are four encoder layers in the pertained BERT-Mini. The two sub-layers of each encoder block are feed-forward and multi-head attention. Self-attention serves as a representation for each of the several heads that make up the multi-head attention. These heads operate in parallel. The self-attention finds the connection between every word in a particular phrase. It is necessary to describe the self-attention process in order to promote greater comprehension. Similarly, although single attention-head averaging hinders this,

multiple-headed attention allows the model to pay to distinct representations and subspaces at different locations. [22]

Contextual embeddings, pre-training on massive unlabeled corpora, bidirectional context modeling, and the transferability of learnt representations to a variety of NLP tasks through fine-tuning are all factors in BERT's success. Following the pre-training stage, the BERT model is subsequently adjusted for certain NLP tasks using its contextual embeddings. In this step, the model's general language understanding is adjusted to the specific task's peculiarities, allowing it to be used in more focused applications. Using labeled data unique to the downstream tasks of interest, BERT is optimized. Sentiment analysis, question answering, named entity identification, and other NLP applications might all be included in these tasks. The model's performance is optimized for the specific needs of the task at hand by adjusting its parameters.

BERT generates and understands language similar to that of a person using a transformer-based neural network. An encoder-only architecture is used by BERT. Both encoder and decoder modules are present in the original Transformer architecture. The choice of an encoder-only architecture in BERT implies that comprehension of input sequences has priority over production of output sequences.

Text is processed sequentially by traditional language models, either from left to right or from right to left. The model's awareness is restricted by this strategy to the immediate context that comes before the target word. Instead of examining the text piece by piece, BERT employs a bi-directional method that takes into account both the left and right context of words in a sentence. This allows BERT to examine every word in a sentence at once.

There are two steps that the BERT model goes through:

- Pre-Training on Large Data

A significant amount of unlabeled text data is used to pre-train BERT. Contextual embeddings, or word representations that consider the context in which they are used in a phrase, are taught to the model. BERT performs a range of unsupervised pre-training tasks.

- Fine-tuning on labeled data

Following pre-training, the model is adjusted to perform well on particular NLP tasks.

Since the purpose of BERT is to produce a language model, only the encoder mechanism is used. Tokens are inserted into the Transformer encoder in a particular sequence. Prior to being processed by the neural network, these tokens are initially integrated into vectors. Contextualized representations are produced as a series of vectors in the output, each of which corresponds to an input token. BERT overlays a categorization layer on top of the encoder's output. The masked words must be predicted using this layer. The vocabulary dimension is created by multiplying the output vectors from the classification layer by the embedding matrix. This stage facilitates the vocabulary space and expected representations' alignment.

A transformer architecture is an encoder-decoder network. It uses self-attention on the encoder side and attention on the decoder side. Similar to the transformer model, BERT's architecture is a multilayer bidirectional transformer encoder.

The model has two architectures as BASE and LARGE. The encoder stack of BERT LARGE has 24 layers, compared to 12 layers for BERT BASE. Additionally, features

of BERTs' architectures are 768 and 1024 hidden units for BASE and LARGE also 12 and 16 attention heads respectively. While BERT LARGE has 340M parameters besides BERT BASE has 110M parameters. The CLS token is the first input used in this model, and then a sequence of text. A classification token is CLS. The input is then sent to the layers. After applying self-attention, each layer transfers the output to the following encoder via a feedforward network. The vector with hidden size is the model's output. The CLS token is represented by the classifier's output from this model.

During training, BERT's primary goal is to correctly identify these hidden words. Predicting the words that are not hidden doesn't really matter to it. This is due to the fact that the actual difficulty is in filling in the blanks, and this tactic aids BERT in developing exceptionally strong word meaning and context comprehension.

4.3 RoBERTa Architecture

The RoBERTa model which based on Google's BERT model was proposed in 2018 by Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov [23]. Significant changes have been made to BERT in the architecture of RoBERTa, especially in the training phase. This involves adding more data and conducting a more thorough pre-training, which leads to outperforming BERT on various metrics. The RoBERTa model is a reimplementation of BERT with some modifications to the key hyperparameters and tiny embedding tweaks.

The transformer architecture with multiple layers is used by BERT. Self-attention heads and concealed dimensions are used in every block. BERT employs next sentence prediction and masked language modeling as its two pretraining goals. In MLM, the special token is substituted for a random sample of the input sequence's tokens. Predicting the masked tokens with a cross-entropy loss is the MLM's goal. Although in actuality data is replicated, so the mask is not always the same for every training phrase, random masking and replacement is done once at the beginning and saved for the duration of training.

In Bert the Next Sentence Prediction predicts whether two segments in the source text will follow one another based on a binary classification loss. Concurrent sentences from the text corpus are used to produce positive examples. Segments from various documents are paired to produce negative instances. Samples of both positive and negative instances have the same likelihood. The NSP goal was created to enhance performance on tasks that come after, like NLP, where thinking about the connections between phrase pairs is necessary. Large mini-batches, a larger byte-level BPE, and dynamic masking FULL-SENTENCES without NSP loss are used to train RoBERTa. In addition, compared to BERT, it is trained with an eight-fold bigger batch size for half as many optimization steps, resulting in four times as many sequences seen in pretraining.

RoBERTa model is trained on a massive dataset of over 160GB of uncompressed text instead of the 16GB dataset used to train BERT model. Built to overcome the shortcomings of BERT, RoBERTa produces cutting-edge results for a variety of Natural Language Processing applications.

Despite its advancements, RoBERTa is not without challenges. Tokenization issues, particularly in languages with complex morphology, and the resource-intensive nature of both training and fine-tuning processes remain concerns. The applicability of RoBERTa, while surpassing BERT in many benchmarks, may not always yield superior results in specific areas or specialized tasks.

The fact that RoBERTa was trained on a larger dataset and with a more efficient training process is one of the main distinctions between it and BERT. Specifically, 160GB of text more than ten times the size of the dataset used to train BERT were used to train RoBERTa.

RoBERTa has almost similar architecture to BERT, but to improve the results some change has been made which they are as follows:

- Removing the Next Sentence Prediction objective;

The model is trained via an auxiliary Next Sentence Prediction loss in the next sentence prediction to forecast whether the observed document segments originate from the same or other texts.

- Training with bigger batch sizes and longer sequences;
- Dynamically changing the masking pattern;

Each time data is passed to the model, different masking is created in RoBERTa.

Although it incorporates changes to the training objectives and hyperparameters, RoBERTa uses a comparable architecture. It depends on the particular deep learning framework being utilized. An outline of the RoBERTa model implementation algorithm is provided below.

- Input Representation:
 - Tokenize the input text using a model-specific tokenizer into words or subwords. Add unique tokens for classification tasks, such as SEP and CLS.
- Embedding Layer:
 - Using an embedding layer, map the input tokens to embeddings. Add positional encoding to the embeddings for capturing the sequence order information.
- Transformer Encoder Blocks:
 - A number of Transformer encoder blocks make up the model's core. Multi-head self-attention and feedforward neural network layers make up each block. For every sub-layer, apply residual connections and layer normalization.
- Training Objectives:
 - The model is trained similarly to BERT to predict randomly masked tokens by using MLM in the input sequence. The NSP objective which is used in BERT removed.
- Optimization:
 - For updating the parameters of the model, using optimizer like Adam during training part.
- Batching and Training:
 - Use the prepared batches of input sequences to train the model, modifying its parameters through optimization and backpropagation.
- Hyperparameter Tuning:

- To maximize performance, adjust hyperparameters like learning rate, batch size, and dropout rates.
- Evaluation:
 - Analyze the model's performance using the metrics such as accuracy, F1 score, precision and recall on validation or test datasets.

Though strategically enhanced, RoBERTa takes over BERT's strong pre-training approach. The capacity for contextual understanding and generalization is strengthened by the addition of additional training data and longer training times. Importantly, RoBERTa performs exceptionally well in tasks like as textual entailment and sentiment analysis. It is not without flaws, though, such as tokenization issues that it faces, especially in languages with complex morphology. Additionally, both the training and fine-tuning processes require a lot of resources, which is a worry. This highlights the complex nature of RoBERTa's applicability: although exceeding BERT in numerous benchmarks, it may not always yield superior results in particular areas or specialized jobs [24].

As a result, considering all the mentioned parameters for analyzing sentiment, it was decided to use RoBERTa-base's model on three different datasets. The first one is 50,000 movie reviews of IMDB and the second one is 100,000 tweets of Sentiment140 dataset and the last one is 42656 reviews of Disneyland Reviews dataset.

Chapter 5

EXPERIMENTAL RESULTS

5.1 Selected Model for Experimental Results

Training a deep network with a lot of data was not possible due to the unavailability of huge data sets and the difficulties of training such a dataset; So, this work requires a lot of processing power and access to this type of data. This is where the usage of pre-trained models plays a key role. In this case, it is enough to allocate a network for experiment.

Although, the advantage of CNN is that it can extract high-level features using convolutional and max-pooling layers, it can not efficiently learn a sequence of correlations. Meanwhile, Bidirectional RNN uses two directions to improve extracting long-term dependencies; but it cannot extract local features in parallel. Firstly, regarding to solve this kind of problems Transformer based model has been chosen to be done like BERT. This model needs a huge dataset but due to limited hardware and GPU issues, datasets with small amount of data have utilized. The BERT model as Transformer-based is the computational resources needed to fine-tune, so it has faced an overfitting problem on small datasets [6].

RoBERTa has two popular models as base and large which both specifications are shown in the Figure 5:

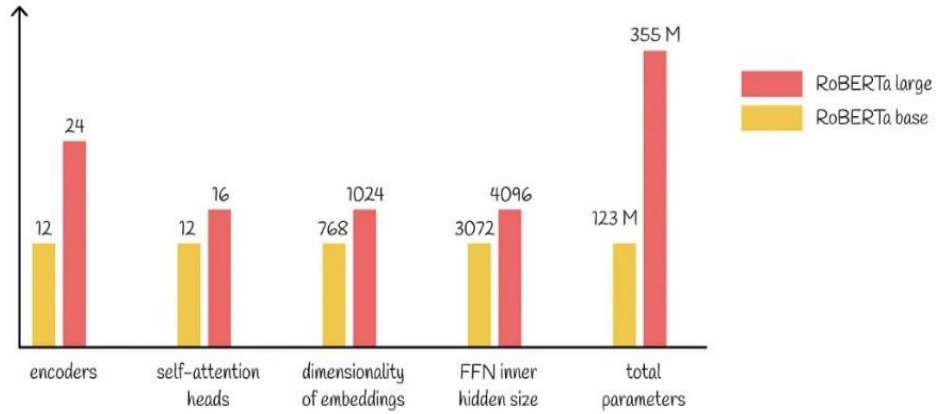


Figure:5 Comparison Between RoBERTa Base and Large

RoBERTa base refers to the smaller variant, with fewer parameters and a smaller architecture, making it computationally more efficient. The choice between RoBERTa base and large depends on the available computational resources. So, in this experiment base model has been used.

5.2 Datasets

The detailed information to represent all used datasets are given in the following.

5.2.1 IMDB Dataset of 50k Movie Reviews

One of the best-known ways of measuring the performance of sentiment analysis algorithms is the IMDB dataset which is a collection of 50,000 movie reviews from the IMDB movie database. Reviews are randomly categorised as 25,000 positive and 25,000 negative ones; thus, the experimental algorithm must classify these correctly.

5.2.2 Sentiment140 Dataset with 1.6 Million Tweets

This is the sentiment140 dataset that includes 1,600,000 tweets extracted using the twitter application, but in this thesis randomly 100,000 tweets were imported into the model. The tweets have divided equally into positive and negative which they have been used to detect sentiment.

5.2.3 Disneyland Reviews

This dataset includes 42656 reviews and ratings of 3 Disneyland branches - California, Hong Kong and Paris. 34125 reviews for training and 8531 reviews to test the dataset.

5.3 Experimental Setup

The experimental categorized into following two parts:

5.3.1 Software Environment

The software tools and libraries used for implementing the experiment:

- Anaconda Platform (version 23.7.4)
- Python (version 3.11.5)
- Spyder (version 5.4.3)
- Libraries:
- Transformers (version 4.23.1)
- PyTorch (version 2.1.1)
- Scikit-learn (version 1.3.0)
- Numpy (1.24.3)
- Pandas (2.0.3)
- Matplotlib (3.7.2)

5.3.2 Hardware Configuration

The hardware specifications of the system used for experimentation:

- CPU model and specifications: 11th Gen Intel(R) Core (TM) i5-11400H @ 2.70GHz 2.69 GHz
- RAM size: 16.0 GB
- Graphic card model and specifications: NVIDIA GeForce RTX 3050 Ti Laptop GPU

5.4 The Metrics Used for Evaluation

Regarding to training dataset, the experimental model is trained by using the pre-trained RoBERTa-base model. The test process was run using the trained model, and the experiment's outcome was obtained. The evaluation results for sentiment analysis, includes accuracy, precision, recall, and F1-score for the datasets separately. While there were some differences in how well this model performed across these metrics which will be presented in the following.

5.5 Algorithmic Steps for Sentiment Analysis by RoBERTa

- Import the needed libraries
- Import data from the dataset
- Encoding the positive and negative labels as numerical values (1 and 0 respectively)
- Preprocessing:
 - Tokenization: Break down the input text
 - Padding: Ensure that all input sequences have the same length by adding padding tokens
- Define some important variables such as batch size and max length
- Split data to train and test as 80 and 20 percent respectively
- Defining Pre-trained RoBERTa Model
- Create the loss function and optimizer
- Train data for 10 epochs and check training process with loss function and accuracy values
- After training model, evaluate it with test data and get the final result of metrics including accuracy, recall, precision, f1 score, and fusion matrix.

5.6 Results

In the following, the results of training and test the model on each dataset will be displayed separately on the charts. They include training accuracy and loss function per epoch, and fusion matrix.

5.6.1 IMDB Dataset Results

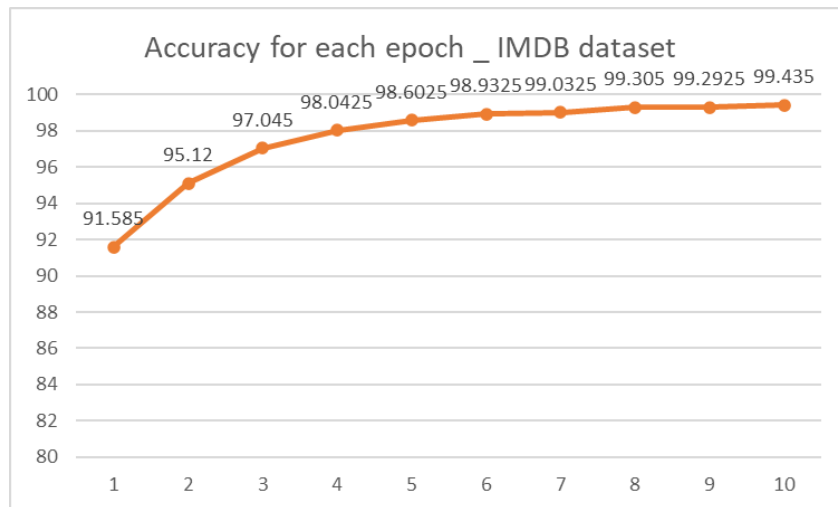


Figure 6: Representation of Training Accuracy on IMDB

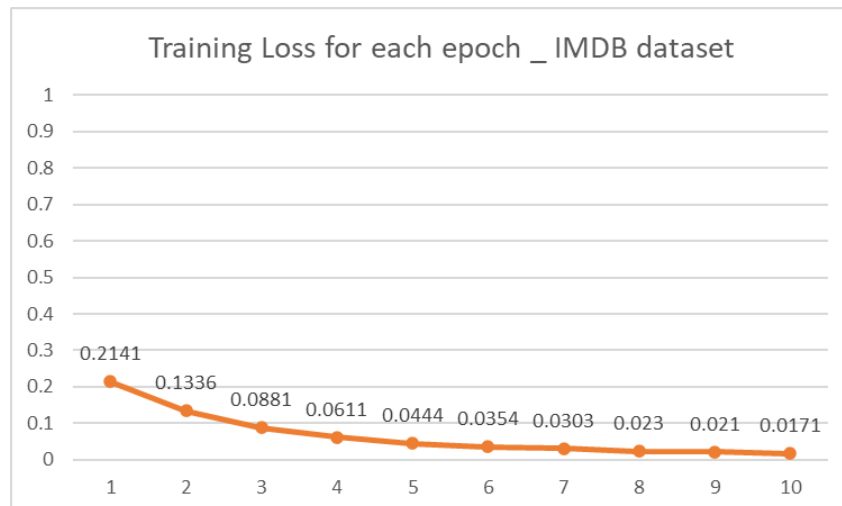


Figure 7: Representation of Training Loss Function on IMDB

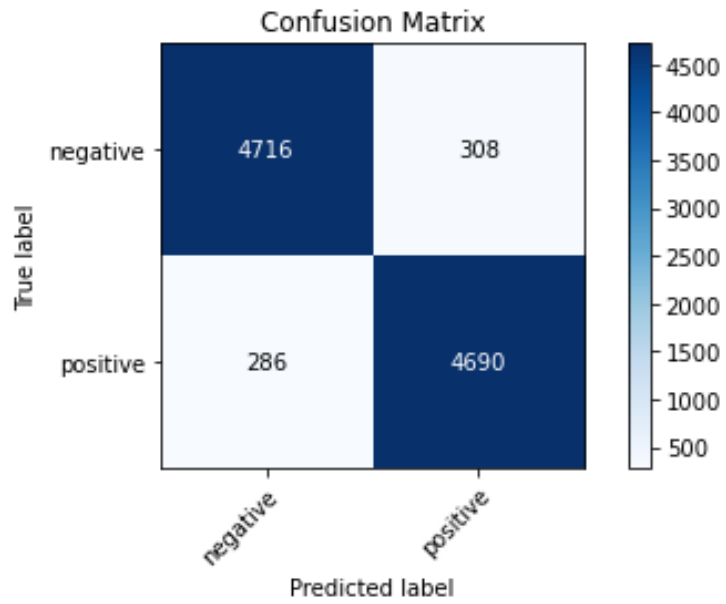


Figure 8: Confusion Matrix on IMDB

5.6.2 Twitter Dataset Results

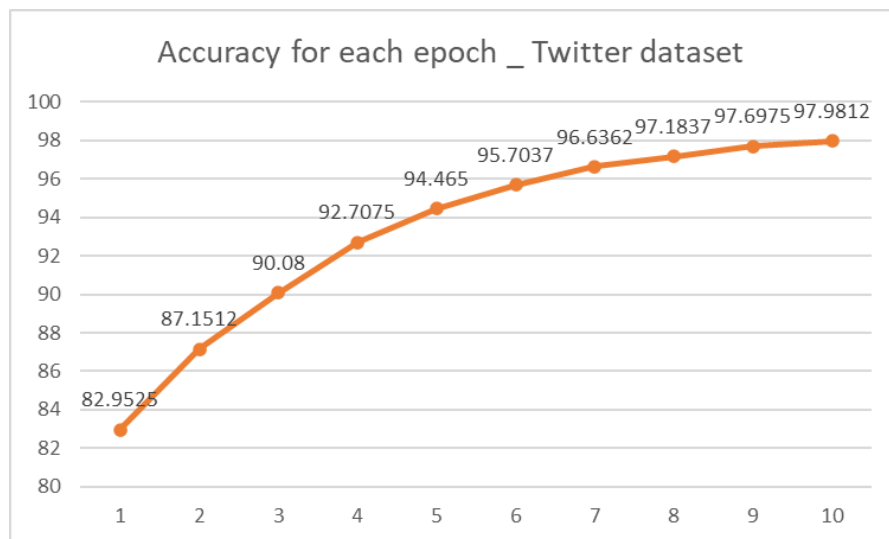


Figure 9: Representation of Training Accuracy on Twitter

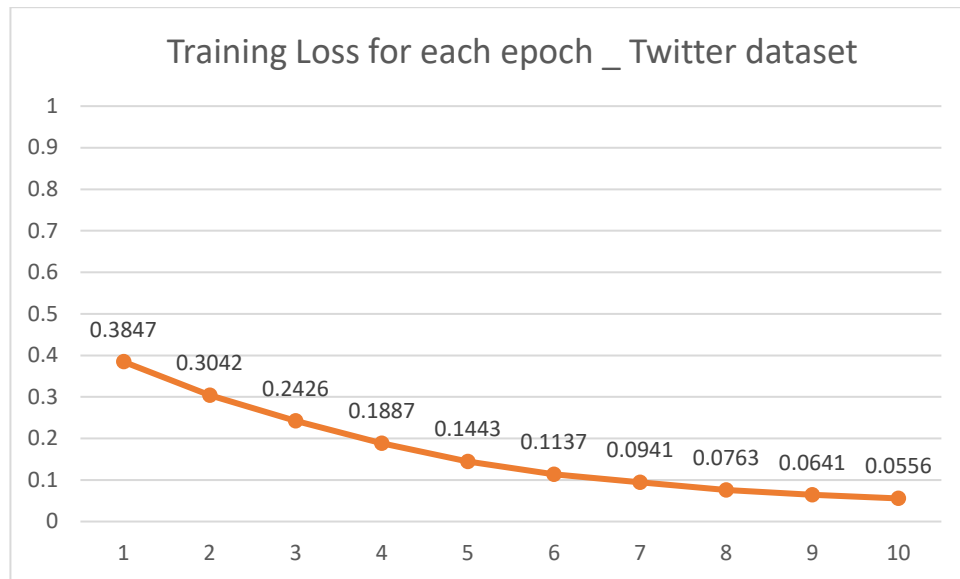


Figure 10: Representation of Training Loss Function on Twitter

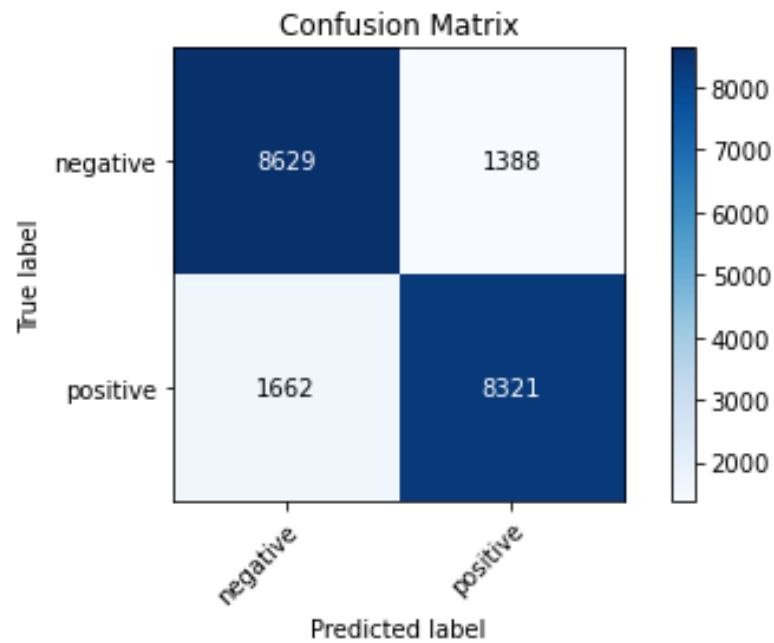


Figure 11: Confusion Matrix on Twitter

5.6.3 Disneyland Reviews Dataset Results

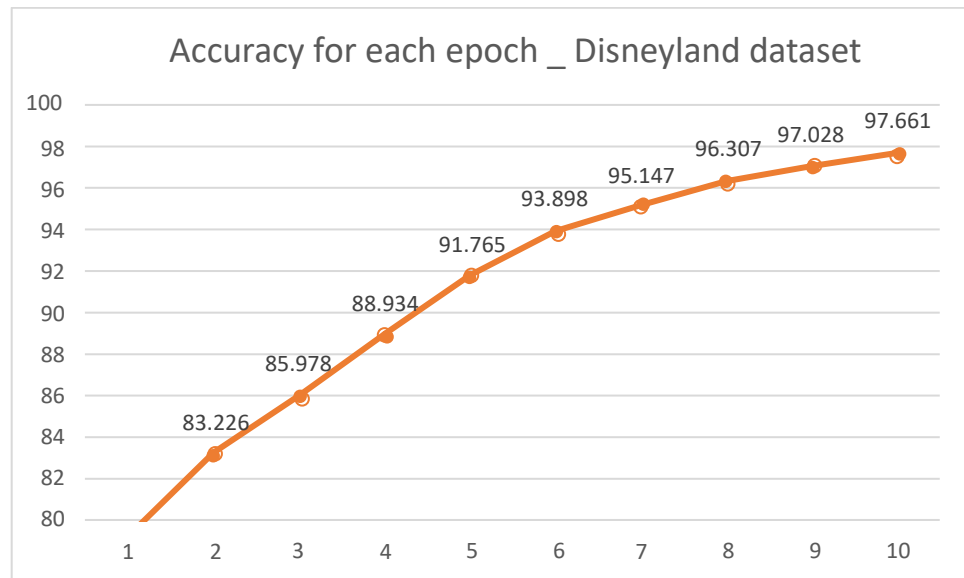


Figure 12: Representation of Training Accuracy on Disneyland Reviews

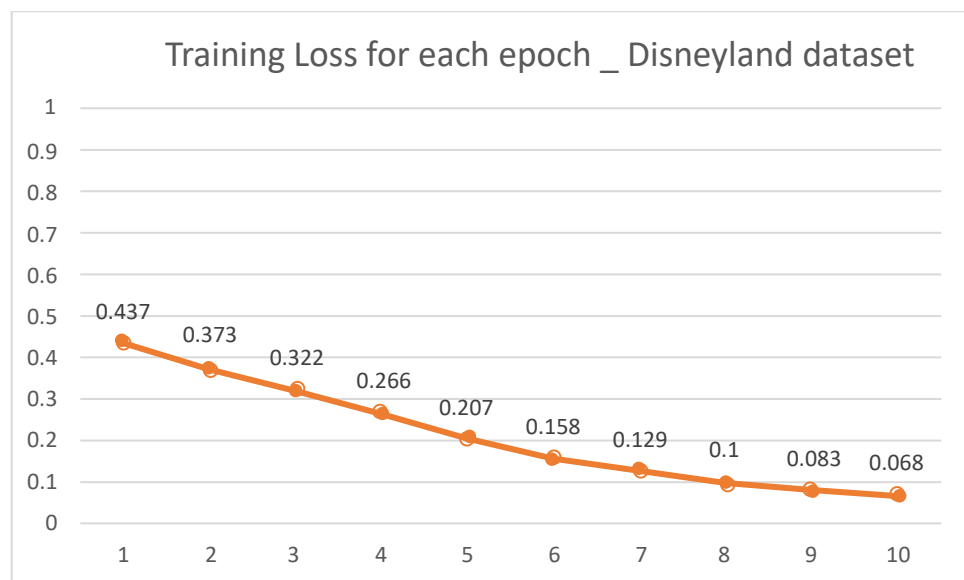


Figure 13: Representation of Training Loss Function on Disneyland Reviews

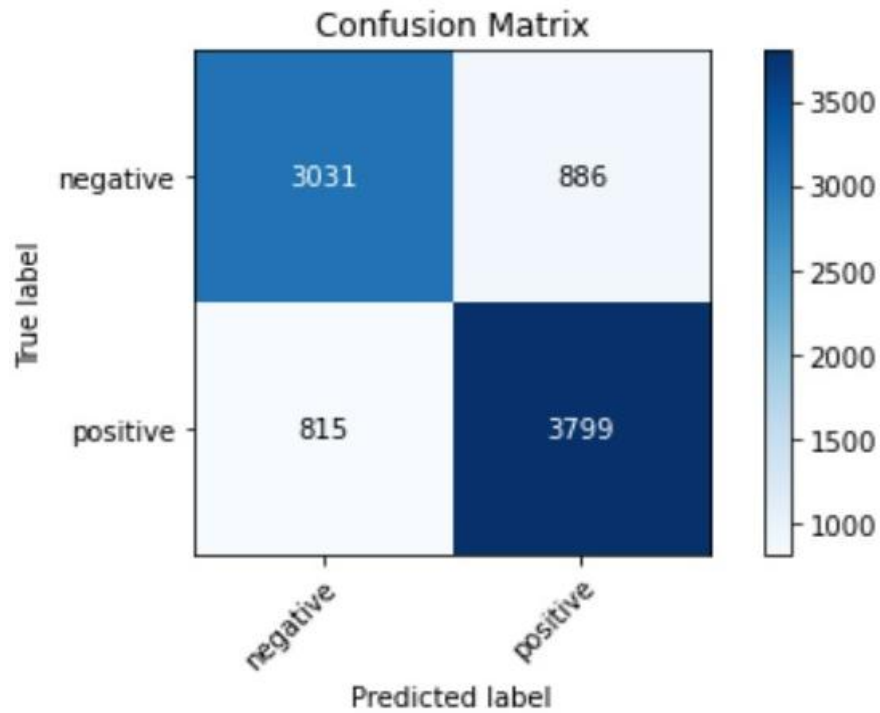


Figure 14: Confusion Matrix on Disneyland Reviews

5.6.4 Discussion

The following table summarizes the performance metrics for sentiment analysis using the RoBERTa model on two distinct datasets: IMDB and Sentiment140. The metrics include Test Accuracy, Precision, Recall, and F1-score. These results showcase the model's effectiveness in analyzing sentiments across diverse datasets.

Table 1: Test results of all used datasets

	Test Accuracy	Precision	Recall	F1-score
IMDB	94.06	0.9406	0.9406	0.9406
Sentiment140	84.75	0.8477	0.8475	0.8474
Disneyland	80.06	0.8004	0.8006	0.8004

In IMDB Dataset the RoBERTa model achieved a high accuracy of 94.06% which this indicates that the model made correct predictions for the majority of the movie reviews. Precision is also high at 0.9406, so when the model predicted a sentiment either positive or negative, it was accurate in its prediction. The recall score is indicated that the model effectively captured a high percentage of actual positive and negative sentiments in this dataset. The F1-score, which considers both precision and recall, is also 0.9406 that is presented the balance performance.

The RoBERTa model achieved a test accuracy of 84.75% on the Sentiment140 dataset. While slightly lower than the IMDB dataset, it still indicates a reasonable level of accuracy.

The RoBERTa model achieved a test accuracy of 84.75% on the Sentiment140 dataset. While slightly lower than the IMDB dataset, it still indicates a reasonable level of accuracy. Precision at 0.8477 suggests that the model was accurate in its predictions of sentiment on this dataset; also, the recall score is 0.8475 that is indicated that the model effectively captured a significant portion of actual positive and negative sentiments. The 0.8474 percentage of F1-score reflects a balanced performance as well as the previous dataset.

On all datasets, the RoBERTa model shows good sentiment analysis capabilities. The IMDB dataset performs better than other movie review datasets, likely because its content is more specialized chosen. Even with the Sentiment140 dataset collected from tweets, the model continues to produce strong results, demonstrating its adaptability to various textual data sources. These measures highlight the RoBERTa model's

efficacy in sentiment analysis across several datasets and demonstrate its potential for real-world use in textual data sentiment classification and understanding.

5.6.5 Comparison the Same Datasets with Different Models

For a better comparison of the model used in this thesis, the models used the same datasets by other articles were also reviewed, and their exact values are shown in the following tables.

Table 2: Comparison other methods with IMDB dataset [25]

Network	Accuracy	Network parameters	Embedding (Tokens)
CNN	0.890	CNN layer (number of filters=64, filter length=5, activation function='relu')	One-hot (1000)
CNN	0.844	CNN layer (number of filters=64, filter length=5, activation function='relu')	WE (500)
CNN+ LSTM	0.891	CNN layer (number of filters=64, filter length=5, activation function='relu'), LSTM layer (activation function='tanh')	WE-tuned (1000)
Proposed Model	94.06	LayerNorm (768, eps=1e-05, activation function='tanh')	Roberta-base (truncation=True, do_lower_case=True)

Table 3: Comparison other methods with sentiment140 dataset [25]

Network	Accuracy	Network parameters	Embedding (Tokens)
LSTM	0.715	LSTM layer (activation function='tanh')	One-hot (300)
CNN+LSTM	0.670	CNN layer (number of filters=64, filter length=5, activation function='relu') , LSTM layer (activation function='tanh')	WE (500)
LSTM	0.713	LSTM layer (activation function='tanh')	WE-tuned (100)
Proposed Model	84.75	LayerNorm (768, eps=1e-05, activation function='tanh')	Roberta-base (truncation=True, do_lower_case=True)

Table 4: Comparison other methods with Disneyland Reviews dataset [26]

	Hong Kong Location		California Location		Paris Location		Proposed Model
	Naïve Bayes	SVM	Naïve Bayes	SVM	Naïve Bayes	SVM	RoBERTa
Accuracy	0.8077	0.8257	0.8538	0.8769	0.7337	0.8059	80.06
Recall	0.8077	0.8257	0.8538	0.8769	0.7337	0.8059	0.8004
Precision	0.6777	0.7839	0.8029	0.8354	0.7174	0.7766	0.8006
F1-score	0.7235	0.7895	0.7883	0.8468	0.6387	0.7850	0.8004

Chapter 6

CONCLUSION

In this thesis, the journey through the realms of Sentiment Analysis has shown. The exploration of emotions as negative and positive categories in three different textual datasets has represented by utilizing advanced model such as RoBERTa.

Getting know the evolution of sentiment analysis techniques is explained in the literature review part, from early rule-based approaches to the advent of deep learning, with particular emphasis on the transformative impact of transformer-based models. The emphasis on transformer-based models, specifically RoBERTa, is showcased their prowess in capturing contextual dependencies and handling vast amounts of data.

The choice of RoBERTa, particularly the base model, was driven by considerations of computational efficiency. Two diverse datasets, IMDB movie reviews and Sentiment140 tweets, served as testing grounds, demonstrating the highly adaptability of the model in distinct contexts.

To conclude, in order to achieve stronger marketing, checking and analyzing users' opinions in the digital world has become a basic need. Therefore, the use of more up-to-date techniques with a higher percentage of accuracy will create targeted marketing and success in sales and businesses in the real world. This thesis contributes to the evolving discourse on sentiment analysis within the larger field of NLP. The exploration of advanced models like RoBERTa lays the foundation for future

endeavors in refining sentiment analysis techniques and addressing the ever-growing challenges in understanding the human language. As NLP continues to evolve, the prospects of enhanced machine learning and adaptive learning from diverse linguistic contexts become increasingly promising.

REFERENCES

- [1] Prabowo, R., & Thelwall, M. (2009). Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2), 143-157.
- [2] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4), 1093-1113.
- [3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- [4] Ahmad, M., Aftab, S., & Ali, I. (2017). Sentiment analysis of tweets using svm. *Int. J. Comput. Appl*, 177(5), 25-29.
- [5] Ahmad, M., Aftab, S., Bashir, M. S., & Hameed, N. (2018). Sentiment analysis using SVM: a systematic literature review. *International Journal of Advanced Computer Science and Applications*, 9(2).
- [6] Parveen, H., & Pandey, S. (2016, July). Sentiment analysis on Twitter Data-set using Naive Bayes algorithm. In *2016 2nd international conference on applied and theoretical computing and communication technology (iCATccT)* (pp. 416-419). IEEE.
- [7] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

- [8] Liao, S., Wang, J., Yu, R., Sato, K., & Cheng, Z. (2017). CNN for situations understanding based on sentiment analysis of twitter data. *Procedia computer science*, 111, 376-381.
- [9] Ce, P., & Tie, B. (2020). An analysis method for interpretability of CNN text classification model. *Future Internet*, 12(12), 228.
- [10] Kurniasari, L., & Setyanto, A. (2020, February). Sentiment analysis using recurrent neural network. In *Journal of Physics: Conference Series* (Vol. 1471, No. 1, p. 012018). IOP Publishing.
- [11] Murthy, G. S. N., Allu, S. R., Andhavarapu, B., Bagadi, M., & Belusonti, M. (2020). Text based sentiment analysis using LSTM. *Int. J. Eng. Res. Tech. Res*, 9(05).
- [12] Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7), 5731-5780.
- [13] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- [14] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

- [15] Banerjee, I., Ling, Y., Chen, M. C., Hasan, S. A., Langlotz, C. P., Moradzadeh, N., ... & Lungren, M. P. (2019). Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artificial intelligence in medicine*, 97, 79-88.
- [16] Widiastuti, N. I. (2019, November). Convolution neural network for text mining and natural language processing. In *IOP Conference Series: Materials Science and Engineering* (Vol. 662, No. 5, p. 052010). IOP Publishing.
- [17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [18] Garg, S., Tsipras, D., Liang, P. S., & Valiant, G. (2022). What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35, 30583-30598.
- [19] Choi, S. R., & Lee, M. (2023). Transformer architecture and attention mechanisms in genome data analysis: a comprehensive review. *Biology*, 12(7), 1033.
- [20] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

- [21] Rothman, D. (2021). Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more. *Packt Publishing Ltd*.
- [22] Kokab, S. T., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, 14, 100157.
- [23] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- [24] Greco, C., Testoni, A., & Bernardi, R. (2020, November). Grounding dialogue history: Strengths and weaknesses of pre-trained transformers. In *International Conference of the Italian Association for Artificial Intelligence* (pp. 263-279). Cham: Springer International Publishing.
- [25] Uysal, A. K., & Murphey, Y. L. (2017, August). Sentiment classification: Feature selection based approaches versus deep learning. In *2017 IEEE International Conference on Computer and Information Technology (CIT)* (pp. 23-30). IEEE.
- [26] Pam, C. O., Ong, W. E., & Rahman, A. (2022). Multiple Approaches in Sentiment Analysis on Disneyland Reviews. *Open International Journal of Informatics*, 10(1), 21-36.