KANTIPUR ENGINEERING COLLEGE

(Affiliated to Tribhuvan University)

Dhapakhel, Lalitpur

[Subject Code: CT755]

A MAJOR PROJECT PRE-FINAL REPORT ON

PRODUCT REVIEW CLASSIFIER USING BI-LSTM

Submitted by:

Aman Mishra	**[41408]**
Bishwas Gautam	**[41420]**
Hemant Lekhak	**[41426]**
Jiwan Bhatt	**[41430]**

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

March, 2024

PRODUCT REVIEW CLASSIFIER USING BI-LSTM

Submitted by:

Aman Mishra	**[41408]**
Bishwas Gautam	**[41420]**
Hemant Lekhak	**[41426]**
Jiwan Bhatt	**[41430]**

Supervised by:

Dr. Gajendra Sharma

Associate Professor

Department of Computer Science,KU

A MAJOR PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR IN COMPUTER ENGINEERING

Submitted to:

Department of Computer and Electronics Engineering

Kantipur Engineering College

Dhapakhel, Lalitpur

March, 2024

KANTIPUR ENGINEERING COLLEGE
DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

APPROVAL LETTER

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a project report entitled ”PRODUCT REVIEW CLASSIFIER USING Bi-LSTM” submitted by

Aman Mishra	[41408]
Bishwas Gautam	[41420]
Hemant Lekhak	[41426]
Jiwan Bhatt	[41430]

in partial fulfillment for the degree of Bachelor in Computer Engineering.

....
Supervisor
Dr. Gajendra Sharma
Associate Professor
Department of Computer Science,KU

....
External Examiner

....
Er. Rabindra Khati
Head of Department
Department of Computer and Electronics Engineering

Date: March 4, 2024

ACKNOWLEDGEMENT

We have taken a lot of effort into this project. However, completing this project would not have been possible without the support and guidance of a lot of individuals . We would like to extend our sincere thanks to all of them. We extend our heartfelt gratitude to Dr. Gajendra Sharma, our esteemed Project Supervisor and Associate Professor in the Department of Computer Science and Engineering at Kathmandu University, Dhulikhel, Kavre. His invaluable guidance, unwavering support, and profound insights have been instrumental in steering our project towards successful completion.We express our deepest appreciation for his mentorship, expertise, and encouragement, which have played a pivotal role in shaping the success of our project. Thank you, Dr. Gajendra Sharma, for being an inspiring mentor and a driving force behind our achievements. We are highly indebted to our Project Manager Er. Bishal Thapa for his guidance and supervision. We would like to thank him for providing the necessary information and resources for this project. Moreover, we would like to thank the Department of Computer and Electronics of Kantipur Engineering College for providing us with the suitable facilities and timely checks for the completion of the project. We would like to express our gratitude towards our friends for their kind co-operation and encouragement which help us a lot in completing this project. Our thanks and appreciations also go to our colleague in developing the project. Thank you to all the people who have willingly helped us out with their abilities.

ABSTRACT

The abundance of fake reviews, comparable to a widespread influence, presents a notable challenge within the domain of online product evaluations. Crafted to either boost specific products or tarnish the reputation of others, these deceptive reviews mislead consumers and undermine the integrity of e-businesses. Recognizing the urgent need to tackle this issue on digital platforms, our paper suggests a solution rooted in extensive data generation within the current digital landscape. We developed an AI-driven systems equipped with deep learning models and natural language processing to thoroughly monitor and analyze review data. At the heart of our study is the development of review classifiers based on a deep bi-directional long short-term memory (Bi-LSTM) model. In an innovative approach, we enhance the model's effectiveness by incorporating the Word2Vec concept, embedding words for more nuanced analysis. This improvement becomes crucial in enabling the model to identify intricate patterns within textual data, contributing to the overall success of the review detection framework. It is essential to highlight that our objective is not to portray this model as a definitive solution, but rather as a practical technological tool designed to complement and enhance current methodologies in the domain of detecting artificially generated reviews.

Keywords– Deep learning , Bi-LSTM,Word2Vec, Natural Language processing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI: Artificial Intelligence

BERT: Bidirectional Encoder Representation from Transformers

Bi-LSTM: Bidirectional long short-term Memory

DFD:- Data Flow Diagram

GPT-2: Generative Pretrained Transformer 2

KNN: K-Nearest Neighbors Algorithm

LDA: Linear Discriminant Analysis

ML: Machine Learning

NLP: Natural Language Processing

PCA: Principal Component Analysis

RF: Random Forest

SDLC: Software Development Life Cycle

SVC: Support Vector Classifier

SVM: Support Vector Machine

UML: Unified Modeling Language

CHAPTER 1
INTRODUCTION

1.1 Background

Nowadays online shopping become a trend in the world. People usually buy things online and these online shopping sites have review options for customer feedback. People usually decide to buy or not purchase products on basis of already available reviews from customers. According to some recent studies, it is reported that about 81 percentage of people read and trust the reviews of products before purchasing them. Because of the such prevailed impact of reviews, several online vendors intend to post fake positive reviews about their products while posting fake negative reviews for opponents' products to obtain financial benefits. Fake or bogus reviews are generated by using two methods which are as follows: In the first method, humans are hired to write scripted reviews about a product. In such scenarios, the appointed reviewers may have never seen the original product but still, give paid reviews about it. While in the second approach, some computer-based automated text generation approaches are used to automatically write fake reviews about the products . Conventionally, human-written paid reviews can be written in a limited amount and the company owners can hire spammers in limited numbers for posting fake reviews. However, the great advancement in the area of machine learning and natural language processing (NLP) has introduced such remarkable algorithms which can post unlimited scripted reviews about any product at a fast pace in comparison to humans.

The such false generated information about the products can affect digital marketing and ecommerce in three main ways. Firstly, such bogus reviews may affect the trust of buyers which could result in a severe market failure. Genuine buyers post authentic reviews to communicate their shopping experience which can be positive or negative depending on their level of satisfaction with products. Such reviews bring a valued service to both the marketplace and sellers to assess the quality of their products. Whereas scripted reviews can damage the trustworthiness of buyers like someone who wants to buy a laptop and the reviews are very positive but actually, the laptop is damaged. Such paid product promotions can systematically demean source authenticity and cause the wrong selection of products which in turn causes the loss of both buyer money and trust in the purchased product and its brand. Secondly, such scripted reviews can have an impact on the ranking of products either in a positive manner (if the generated reviews are positive) or in a negative manner (if the bogus reviews are negative) . This is due to the reason that the algorithms employed to rank the online markets utilize these reviews as a measure to estimate the ranking of a product against its competitors. Hence, such generated false content may cause biased competition, in which the ranking of a product is falsely overstated or collapsed. So, such strategies can be used as a weapon by an organization to affect the reputation of its rivals. Lastly, the fake reviews will not only impose a superficial or reputational impact but accompanies a monetary cost as well. Like, a study conducted in shows that a one-star reduction in a brand's Yelp ranking shows a five to nine percent decline in income. Hence, the generation of false reviews is imposing a serious threat to both companies' profit and buyers' well-being. Therefore, automatic spam identification is an essential task because it can protect buyers from a waste of money and time. Even though several studies have been conducted to identify the fake generated reviews, however, this type of spam identification is difficult because spammers can easily hide their identities. So, there is a need for such a method that can detect and filter out the genuine reviews[1].

1.1.1 Importance of User Reviews

Online purchasers on e-commerce sites are increasing day by day. Online purchasers often post reviews/opinions about certain product they have used. In other words, opinions are content created by users on e-commerce websites to express experience of users about any service or product. Importance of user reviews can be viewed from user and business perspective. From user perspective, these reviews can influence new customers/users for purchasing decision of certain product in a good or bad way. Decision of new purchasers is influenced by reviews of users. For purchasing online, user often visit e-commerce sites rich with user experience about products. So, quality and number of user experience can affect user traffic on site.

1.2 Problem Statement

Fake reviews are causing trust issues on online platforms. They deceive consumers and harm businesses' reputations. There is need of a a reliable system that can accurately detect fake reviews and ensure that online reviews are trustworthy, so people can make informed decisions and businesses can maintain their credibility.

1.3 Objectives

i To develop a Fake Review Detection program implemented as a browser extension, aiming to effectively identify artifically generated reviews on online platforms.

1.4 Application Scope

A fake review detection system can be applied in various domains where online reviews play a significant role in influencing consumer decisions. Here are some potential application scopes for a fake review detection system:

i E-commerce platforms: Fake reviews can be a major problem on e-commerce websites where customers rely on reviews to make purchasing decisions. A fake review detection system can help identify and remove fraudulent reviews, ensuring that customers receive accurate and reliable information.

ii Online service marketplaces: Freelance platforms and service marketplaces can utilize fake review detection systems to identify and remove fake or biased reviews. This helps both service providers and customers in making well-informed decisions about hiring or accepting job offers.It can assist in identifying suspicious reviews that may have been fabricated or manipulated, contributing to a more reliable and rigorous peer-review process.

1.5 Features

i Detect the authenticity of reviews based on selected input review.

ii Easy to use extension based system for a user to check authenticity in real life datas.

1.6 System Requirements

This project needs certain hardware and software requirements in order to be developed and run. These requirements are discussed below:

1.6.1 Development Requirements

Development requirements mainly focus on only the building blocks of the application. So, for developing this fake product reviews detection system, we need some requirements for carrying out the different tasks. It is broadly classified into two categories:

Hardware Requirement(Minimum)

Since this application is specifically focused on a specific task. So for this, we needed the most feasible and compatible devices. They are listed below:-

- PC with 8 GB RAM or more.
- i5 processor with 7th generaion or more.

Software Requirement

- Operating System : Windows/ Ubuntu/ macOS

1.6.2 Deployment Requirements

Hardware Requirement(Minimum)

- Laptop/PC

Software Requirement

- Operating System : Windows / Ubuntu/ macOS

1.7 Feasibility Study

The feasibility study is an analysis that we considered all of the relevant factor required for a good and systematic project development which includes economical, technical, operational and schedule considerations.

1.7.1 Economic Feasibility

The economic feasibility of the project is robust, with a well-managed cost structure, efficient resource utilization, and considerations for future scalability. The positive cost-benefit analysis and strategic economic planning underscore the project's viability from an economic perspective.

1.7.2 Technical Feasibility

Throughout the project development, it was determined that the required technical skills for utilizing the application are easily learnable, making the system accessible to a broader audience. This user-friendly approach aligns with the goal of ensuring that users can easily grasp the functionalities of the application without extensive technical expertise.

1.7.3 Operational Feasibility

Through the advancement in technologies, nowadays any sort of software or application is no longer hard to operate. Thus, the user does not need to be expert in using web-based application but with some basic knowledge they can easily operate and make use of our extension. This application highly focuses on security, affordability, usability, maintainability, supportability etc.

1.7.4 Schedule Feasibility

As the project concludes successfully, the adherence to the proposed schedule underscores the feasibility of the timeline set during the planning phase. The disciplined execution of the project timeline reflects the team's commitment to delivering results in a timely manner.

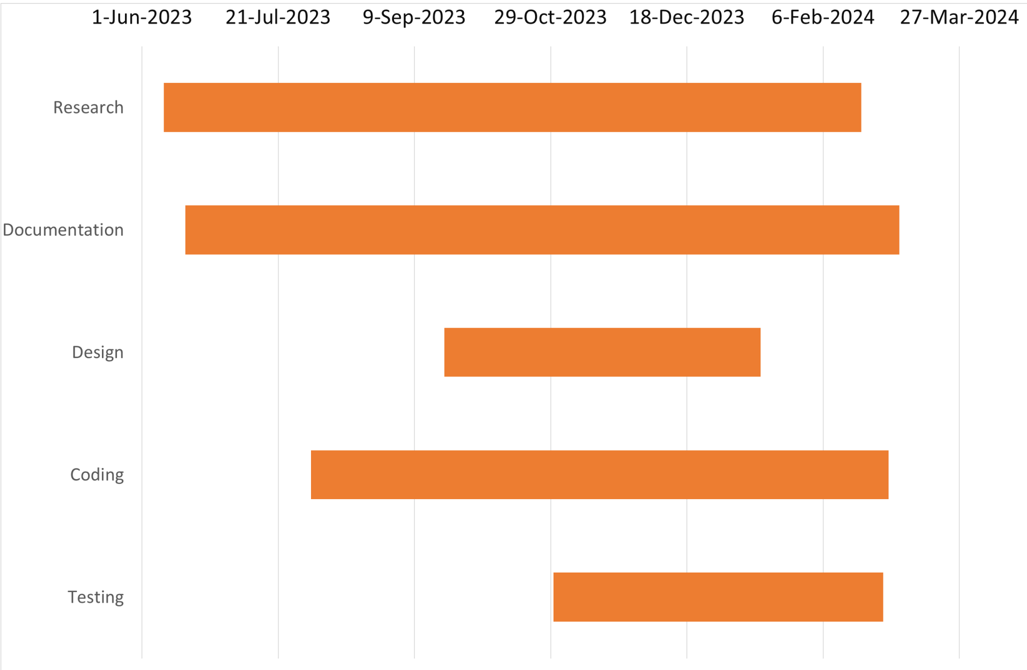

Figure 1.1: Gantt Chart

CHAPTER 2
LITERATURE REVIEW

2.1 Related Project

1. Fakespot in E-commerce:
   Fakespot employs a sophisticated blend of natural language processing (NLP) techniques and machine learning algorithms to detect fake reviews in the e-commerce sector. The process begins with data collection from diverse e-commerce platforms, followed by extensive preprocessing to clean and standardize the textual data. Utilizing sentiment analysis, Fakespot assesses the emotional tone of each review, distinguishing between positive, negative, and neutral sentiments. Anomaly detection techniques, such as clustering and outlier analysis, are then applied to identify reviews exhibiting unusual patterns. The platform leverages advanced machine learning models, including ensemble methods, to analyze linguistic nuances, reviewer behavior, and semantic structures within reviews. By combining sentiment analysis, anomaly detection, and machine learning, Fakespot provides consumers with trustworthiness ratings, offering a comprehensive solution for combating fake reviews and enhancing the reliability of product recommendations in the e-commerce space.
2. TripAdvisor's Content Integrity Team in Travel and Hospitality:
   TripAdvisor's Content Integrity Team employs a meticulous process combining automated tools and natural language processing (NLP) to maintain the authenticity of user-generated reviews in the travel and hospitality industry. The process begins with the collection of a vast corpus of traveler reviews. Automated tools preprocess the reviews, applying techniques such as tokenization and part-of-speech tagging. NLP models analyze the linguistic patterns, coherence, and consistency of language within reviews. The team utilizes sentiment analysis to assess the overall tone of reviews, identifying potential anomalies. Human moderators complement automated processes by conducting manual reviews to identify subtle instances of fake reviews that may elude automated algorithms. This synergistic approach, combining automated NLP techniques with human

expertise, ensures a robust content integrity process on the platform

3. SentiGeek

   SentiGeek specializes in sentiment analysis within the technology sector, employing advanced natural language processing (NLP) techniques to extract valuable insights from customer reviews. The process commences with the collection of a diverse dataset of technology-related reviews. Preprocessing techniques, such as stemming and lemmatization, are applied to prepare the textual data. SentiGeek's NLP models then perform sentiment analysis, categorizing reviews into positive, negative, or neutral sentiments. Named Entity Recognition (NER) is utilized to identify key entities and phrases related to technology products or services. The tool analyzes syntactic and semantic structures within reviews, identifying linguistic patterns indicative of customer satisfaction or dissatisfaction. Through the integration of sentiment analysis, NER, and linguistic pattern recognition, SentiGeek provides technology businesses with actionable insights derived from the language used in customer feedback, facilitating informed decision-making and enhancing overall customer relations.

2.2 Related Research

Many researchers have attempted to propose methods for the timely detection of fake reviews. In this section, we have performed an analysis of existing techniques used for fake reviews detection and classification. One such method was proposed in, where initially, a preprocessing step was performed to clean the data. Then the three-dimensional time-series approach was used to compute the text and behavioral features of online reviews. Finally, the random forest (RF) classifier was trained on the calculated features to classify the real and fake reviews. The approach shows better fake reviews classification performance, however, the technique requires more evaluations to show its robustness.

Adelani et al. presented a sentiment-observable bogus reviews creation and detection approach. Initially, the GPT-2 framework was applied to produce extensive reviews with the preferred sentiment chosen from the target website. In the next step, the BERT text classifier was applied to remove the unwanted sentiments. After this, three approaches namely Grover, GLTR, and GPT-2 classifiers were nominated for bogus review identification. The proposed approach shows an equal error rate of 22.5 percentage for fake reviews detection which requires improvements to better tackle complex real-world scenarios[2].

Another approach was proposed in to detect the fake generated reviews from the websites. In the first step, a preprocessing step was applied to remove the unnecessary details from the input data. Then, the textural features were extracted from the processed data. After this,several ML-based classifiers namely the KNN, Naive Bayes (NB), RF and the SVM classifiers were trained on the computed keypoints to accomplish the classification task. The approach obtains the best results for the KNN classifier, however, lacks to generalize well to real-world scenarios.

Another method for bogus review identification was presented in which the computed features were passed to numerous classifiers namely the SVM, KNN, RF, DT, and LDA to categorize the original and generated reviews. The approach shows the best results for the SVM classifier, however, may not generalize well to unseen cases. Several techniques have been reported by the researchers for the correct identification of bogus reviews, however, several approaches lack to integrate the link between semantics and time of the given reviews. Moreover, the influence of several sources of information for bogus reviews was not taken into count. Such information establishes a complicated, extensive, diverse relation among assessors, reviews, stores, and supplies. To tackle such issues a work was presented in another project where a dynamic knowledge graph-based approach for bogus reviews detection. The model computed four kinds of entities employing a developed NN framework namely the sentence vector/twin-word embedding-based LSTM model. The model works well for the fake reviews detection, however, at the charge of the enhanced computational burden[3].

J. Salminen et al. used two language models to generate fake product reviews. The Amazon dataset was used to compare the performance between ULMFiT and GPT-2. The experiments showed that GPT-2 outperformed ULMFiT on all relevant metrics. Therefore, they used GPT-2 to build the fake review dataset for the classification task. After that, they compared various algorithms and concluded that the RoBERTa model yields the maximum accuracy. The research was also conducted on humans with their accuracy being 53.36 percentage which clearly shows machines outperformed humans on classification task[4].

The paper titled ”Advanced Misinformation Detection: A Bi-LSTM Model Optimized by Genetic Algorithms” addresses the pressing issue of widespread misinformation in the digital landscape, offering a comprehensive solution through AI-driven systems and deep learning models. The proposed methodology centers on developing misinformation classifiers using a potent Bi-LSTM model, further enhanced by the integration of a genetic algorithm (GA) for optimal neural architecture. This research introduces a novel contribution to misinformation detection by presenting a GA-tuned Bi-LSTM model, showcasing superior performance in discerning misinformation. The study goes beyond mere model introduction, including the implementation and evaluation of various machine learning models, serving as benchmarks that highlight the enhanced performance of the proposed model. Additionally, a comprehensive comparative analysis is conducted, evaluating the GA-tuned Bi-LSTM model against both implemented machine learning models and state-of-the-art techniques. The study stands out for its holistic approach, encompassing various performance metrics, thereby providing a balanced evaluation of the proposed model’s effectiveness in addressing the pervasive challenge of misinformation[5].

The paper titled ”Data-Driven Solution to Identify Sentiments from Online Drug Reviews” addresses the significant role of social networking sites in hosting user-generated content, especially regarding medications and health. Given the challenge of extracting valuable insights from extensive comment data on drug experiences, the study employs sentiment analysis (SA) to enhance medication review categorization. The approach involves training five machine learning (ML) algorithms and four deep learning (DL) classifiers on diverse features and word-embedding approaches, respectively. Results showcase the bidirectional LSTM (Bi-LSTM) model, trained on GloVe embedding, outperforming with an accuracy of 97.40 percentage and F1 score of 97.42 percentage. The study aims to develop ML and DL algorithms capable of handling noisy data and subjectivity in online drug reviews for more accurate results. Contributions include a comprehensive comparative analysis of ML and DL algorithms and the provision of a large labeled corpus of drug reviews, offering valuable insights for pharmaceutical companies to enhance products and services. The web application developed as part of this study serves as a practical tool for consumers’ experiences with drugs[6].

CHAPTER 3
METHODOLOGY

3.1 Working Mechanism

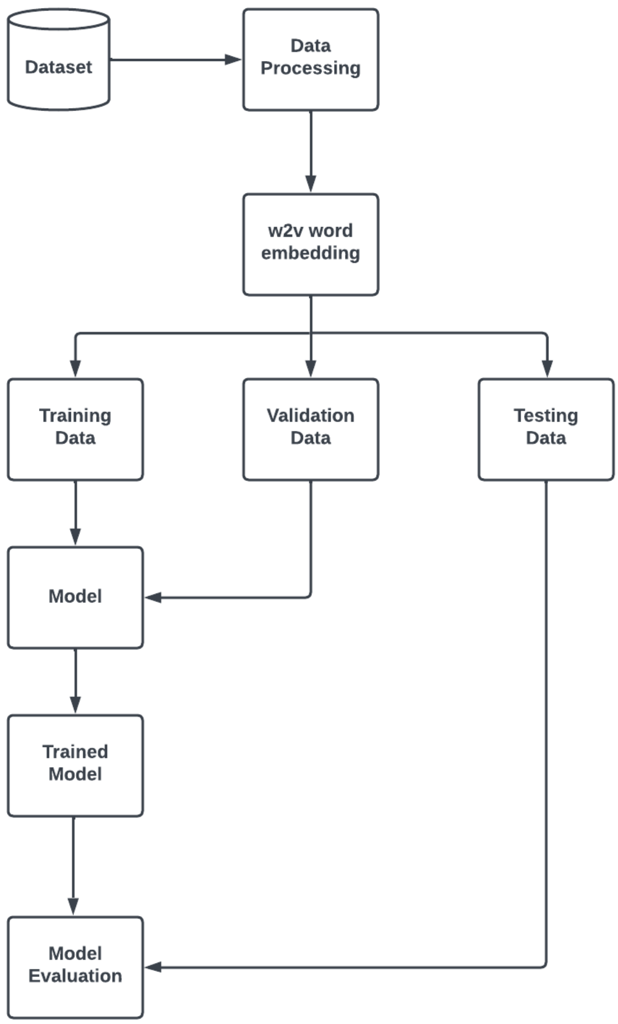

Figure 3.1: System Block Diagram

The components used in our block diagram are as follows:

1. Dataset

   The dataset we collected includes Top-10 Amazon categories with the most product reviews (i.e., Books; Clothing, Shoes, and Jewelry, Home and Kitchen, Electronics, Movies and TV, Sports and Outdoors, Kindle Store, Pet Supplies, Tools and Home Improvement, Toys and Games). Reviews from these categories account for 88.4 percentage of the reviews in the baseline dataset, thereby representing the baseline dataset (Amazon product reviews) reasonably well. The resulting dataset includes 20,000 artificially-generated (fake) reviews. It also includes 20,000 (real) reviews written by humans (i.e., original samples from the Amazon dataset). Hence, there are 40,000 reviews in total. The collected samples also account for proportions as per the original distribution in the sample, for example, if the proportion of 50-word reviews in the Amazon dataset is 0.5 percent of the total reviews, then 0.5 percent of this dataset reviews are also 50 words of length[4].

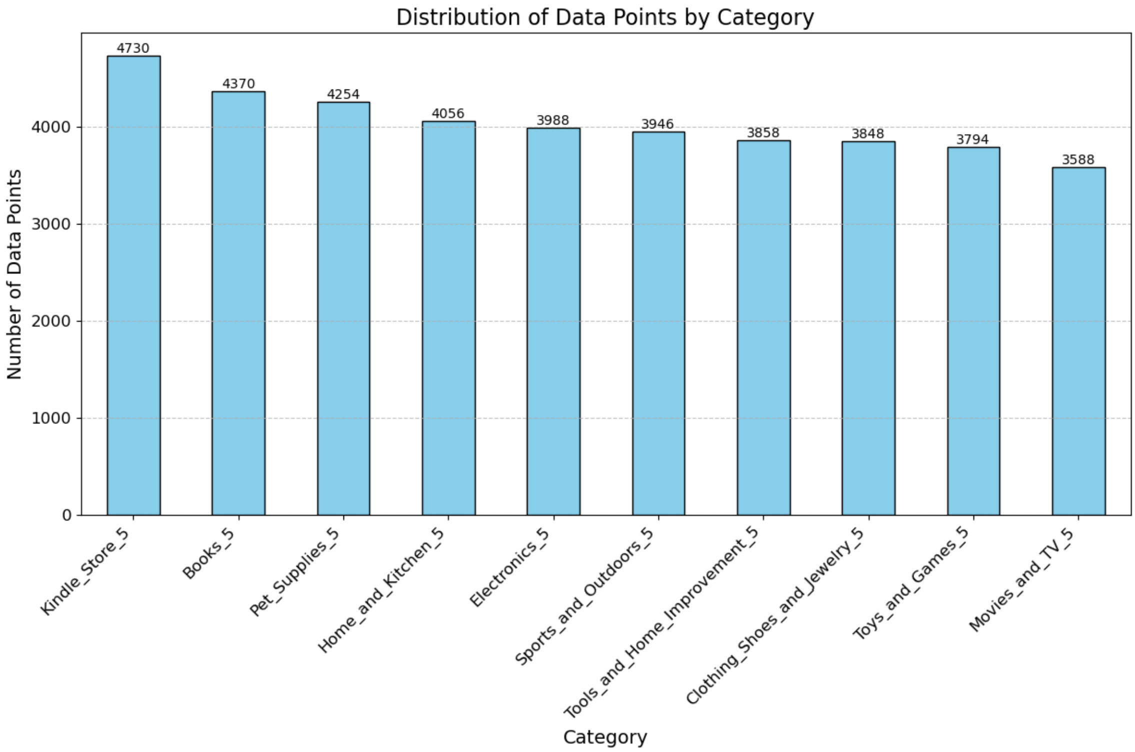

Figure 3.2: Distribution Of Dataset

2. Data Preprocessing:

   Mostly the data has different unwanted phrases or words like misspelling, stop words, etc., which can affect the performance of the technique and increases the computation cost. To solve this problem preprocessing steps which include stop-word removal, spelling correction, tokenization, and noise removal is performed. Some of the important preprocessing steps are:

   - The common stop words in textual data are: “the, and, for, a, about, after”, and do not have any vital impact on classification approaches are removed.
   - The lowercase format is essential for text analysis because it can provide

accurate comparative procedures and saves execution time. For this purpose, the whole textual data is converted into lowercase.

- Tokenization, a vital step in text processing, breaks down the text into smaller units, like words or phrases. For example, consider the review text ”This movie was fantastic!” Tokenization would split it into individual tokens: [”This”, ”movie”, ”was”, ”fantastic”, ”!”] This creates a structured representation for further processing.

  Other preprocessing steps like stemming and lemmatization are also used in text classification. However, in certain cases, stemming and lemmatization can negatively impact text categorization performance.

  Before prepocessing:”Great phone! Super fast, sleek design, and the camera quality is amazing. Definitely recommend it!”.

  After preprocessing :[’great’, ’phone’, ’super’, ’fast’, ’sleek’, ’design’, ’camera’, ’quality’, ’amazing’, ’definitely’, ’recommend’]

3. Word2vec word embedding:

   Word2Vec is a widely used algorithm for learning word embeddings, which are dense vector representations of words in a continuous vector space. The underlying concept of Word2Vec is based on the idea that words appearing in similar contexts tend to have similar meanings. The algorithm operates on large corpora of text, learning vector representations by predicting surrounding words given a target word or predicting a target word given its context.

   There are mainly two methods for learning representations of words:

   - Continuous bag-of-words model:

     Predicts the middle word based on surrounding context words. The context consists of a few words before and after the current (middle) word. This architecture is called a bag-of-words model as the order of words in the context is not important.

   - Continuous skip-gram model:

     Predicts words within a certain range before and after the current word in the same sentence. A worked example of this is given below.

For this project, we employed Word2Vec models for embedding purposes. For this project we trained word2vec using 560,000 Amazon reviews with additional to 40000 reviews from our dataset.

Training parameters for w2v:

Vector Size = 300

Window Size = 5

Minimum Count = 1

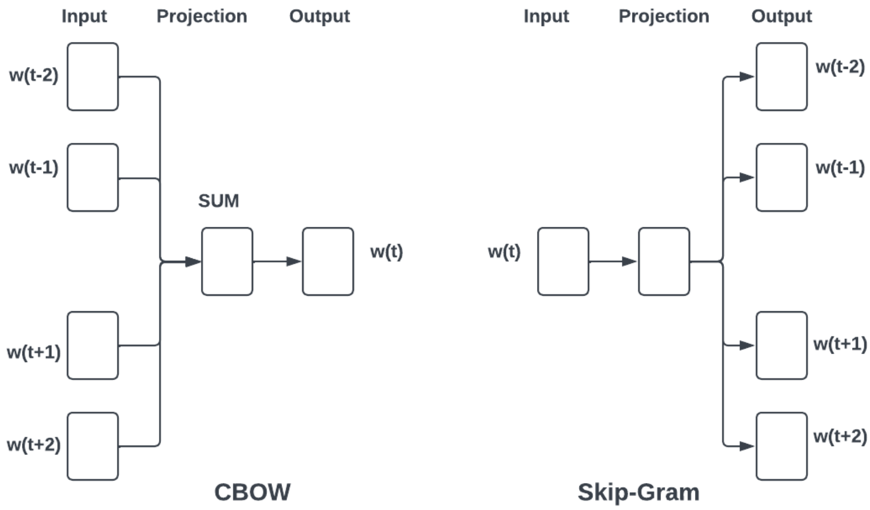

Figure 3.3: Architecture Of word2vec

4. LSTM :

   An LSTM cell is the building block of our proposed model. LSTM is considered an upgraded form of RNN. One of the main limitations of RNNs is the vanishing gradient problem, which occurs when the gradients of the network weights become too small during backpropagation, causing the network to stop learning. LSTM networks solve this problem by introducing gates that regulate the flow of information in the network, allowing it to selectively remember or forget information over long periods. Another advantage of LSTM networks is their ability to capture long-term dependencies in sequential data. Unlike RNNs, which may forget important information over long sequences, LSTMs can remember important information for much longer periods, making them more effective at capturing complex patterns and relationships in sequential data. Figure 2 shows the components of a typical LSTM unit, which consists of a cell, an input gate, an output gate, and a forget gate. The three gates control the flow of information into and out of the cell, and the cell remembers values across arbitrary time intervals. The detailed working mechanism of an LSTM unit can be explained through the elaboration of its three gates: forget gate, input gate and output gate[7].

LSTM has three gates:

- Forget gate:

  The forget gate's primary function is to determine which bits of the cell state are helpful. To achieve this, the neural network is fed both the old hidden state and the fresh input data. Using a sigmoid activation function, the neural network creates a vector with each element falling within the range of [0,1]. The output denoted by f_{t} from the forget gate can be expressed using Equation 3.1. In this equation, σ is the activation function, w_{f} and b_{f} are the weight and bias of the forget gate, respectively. $H_{\text{t-1}}$ and X_{t} represent the concatenation of the last hidden state and the current input, respectively.

$$F_t = \sigma(w_f[H_{t-1}, X_t] + b_f) \tag{3.1}$$

- Input Gate:

  The input gate has two primary goals. The first step is to determine what part of the new data should be used for the cell/memory update. Second, it tries to find out whether the new input data is worth remembering at all. The input gate undergoes two stages to achieve this. These two stages are depicted using the following Equations (3.2) and (3.3). In Equation (3.2), the weight matrices and bias of the operation are represented by w_{c} and b_{c}, respectively. Meanwhile, the activation function used is the hyperbolic tangent (tanh) function. In equation (3.3), w_{i} and b_{i} are the weight matrices and the bias of the input gate, respectively.

$$\widetilde{C}_t = \tanh(w_c[H_{t-1}, X_t] + b_c) \tag{3.2}$$

$$i_t = \sigma(w_i[H_{t-1}, X_t] + b_i) \tag{3.3}$$

Pointwise multiplication is then used to update the cell state given by Equation 3.4.

$$C_t = f_t \odot C_{t-1} + i_t \odot \widetilde{C}_t \tag{3.4}$$

- Output Gate:

  Determining the new hidden state is the output gate's primary function. To produce the filter vector o_t as given in Equation 3.5, the prior hidden state and present input data are passed through the sigmoid activated network.

$$o_t = \sigma(w_o[H_{t-1}, X_t] + b_o) \tag{3.5}$$

The filter vector is then applied to the cell state by pointwise multiplication after being passed through a activation function that compresses the values into the range [-1, 1]. Equation 3.6 illustrates the creation and output of a new hidden state along with a new cell state.

$$H_t = o_t \odot \tanh(C_t) \tag{3.6}$$

The new hidden H_t state becomes a prior hidden state H_{t-1} to the following LSTM unit, whereas the new cell state C_t becomes a previous cell state C_{t-1}.

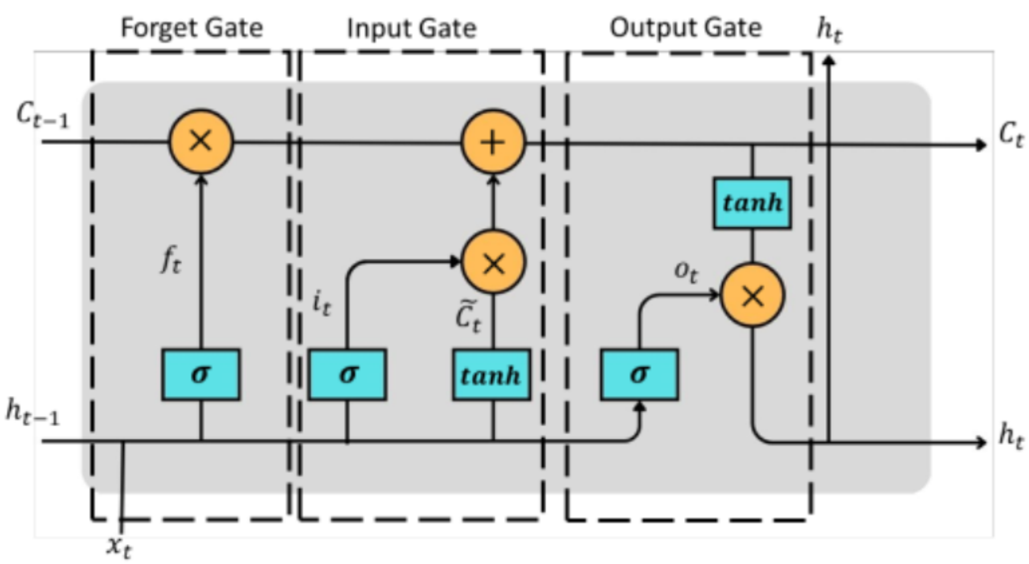

Figure 3.4: Architectural view of LSTM cell

5. Bi-LSTM

An LSTM layer is made up of a sequence of LSTM cells, and the input data is processed only in a forward direction. On the other hand, Bi-LSTM includes an additional LSTM layer that processes the data in a backward direction, as shown in Figure 3. Training a Bi-LSTM network is equivalent to training two separate unidirectional LSTM networks. One of these networks is trained on the original input sequence, while the other is trained on a reversed copy of the input sequence. This approach provides the network with more contextual information, leading to faster and more comprehensive learning of the problem.

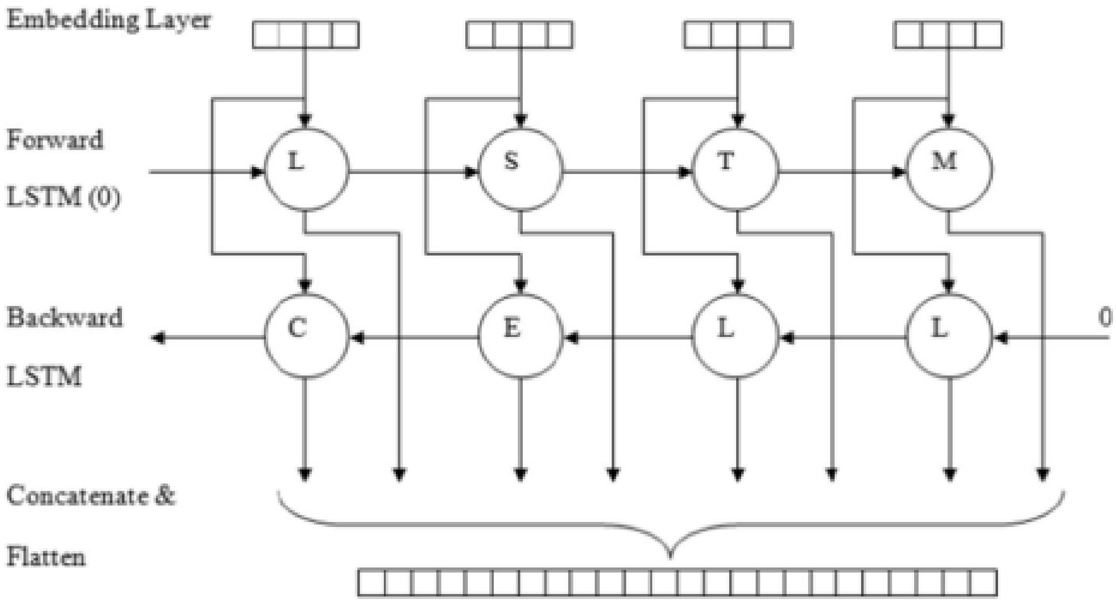

Figure 3.5: Visual depiction of the Bi-LSTM structure

6. Model Architecture

The preprocessed review is initially passed through an embedding layer, producing a 300-dimensional embedding output. This embedding output is then fed into a bidirectional LSTM layer with 100 units, allowing it to capture context from both forward and backward sequences. The bidirectional LSTM enhances the model's ability to comprehend dependencies in the input sequence. The output from the bidirectional LSTM is then directed to a dense layer with 80 units, featuring a Rectified Linear Unit (ReLU) activation function, a dropout layer, and L2 regularization to prevent overfitting. This dense layer serves as a feature extractor, abstracting and compressing the learned information.

Following the initial dense layer, the output proceeds to another bidirectional

LSTM layer with 100 units, accompanied by a dropout layer. This additional bidirectional layer further refines the model's understanding of sequence patterns. The subsequent dense layer, also with 80 units, dropout, and L2 regularization, continues to refine the feature representation. The final output from this dense layer is then directed to the output layer, which employs a sigmoid activation function for binary classification.

Upon compilation, the model utilizes the Adam optimizer with a variable learning rate and clip value. The 'binary cross-entropy' function is employed as the loss function for calculating the model's loss during training.

The model was experimented with a similar architecture but different sets of hyperparameters until we get the best model.

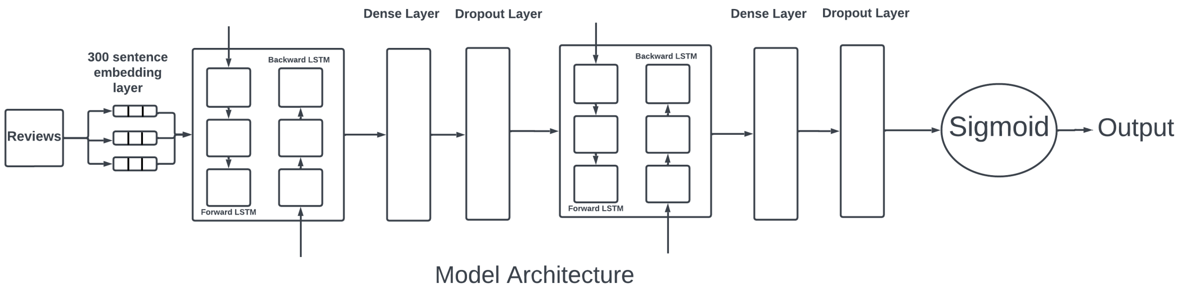

Figure 3.6: Architecture Of Classification Model

- Hyperparameters
  (a) Learning rate

  This hyperparameter controls the step size the optimizer takes when updating the model's weights during training. A smaller learning rate often leads to slower but more stable convergence, while a larger learning rate can lead to faster convergence but also increase the risk of the model overshooting the minimum.

  Learning rate: 0.0007

  (b) Epochs

  This hyperparameter refers to the number of times the training algorithm iterates over the entire training dataset.

  Epochs: 70

  (c) Batchsize

This hyperparameter determines the number of data samples the model processes at a time during training. Smaller batch sizes can lead to more frequent updates and potentially smoother convergence, while larger batches can leverage vectorized operations and improve training speed.

Batch size: 64

(d) Dropout

Dropout is a technique used to prevent overfitting in neural networks. During training, a random subset of neurons (along with their incoming and outgoing connections) is temporarily ignored at each training step .This helps prevent the network from relying too heavily on any specific features or becoming overly sensitive to the training data.

Dropout: 0.4(In Both Dense Layer)

Dropout: 0.4(In Both Dense Layer)

(e) Clipvalue

Gradient clipping is a technique used to address the issue of exploding gradients, which can occur during training and lead to unstable learning. It essentially sets a threshold for the magnitude of the gradients, and any gradients exceeding that value are clipped to the threshold.

Clip value: 0.75

(f) L2 regularization

It is a technique to mitigate the risk of overfitting by adding a penalty term to the loss function. Specifically, it discourages the model from assigning excessively large weights to the features.

L2 regularization: 0.01(In Dense layer only)

3.2 System Diagrams

3.2.1 Use Case Diagram

A use case diagram is graphical illustration of interactions along the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize the system requirements. In this context the term system refers to something being developed. Use case diagram are employed in UML, a standard notation for modelling of real-world objects.

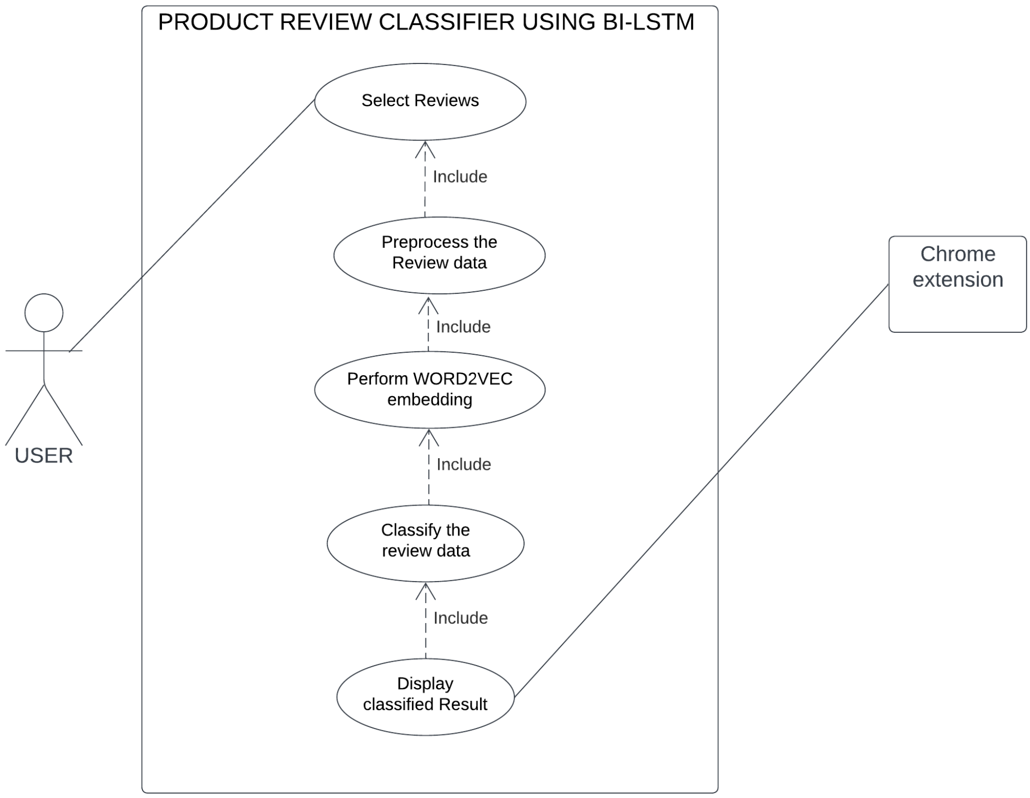

Figure 3.7: Use Case Diagram

3.2.2 Data Flow Diagram

Figure 3.8: DFD Level 0

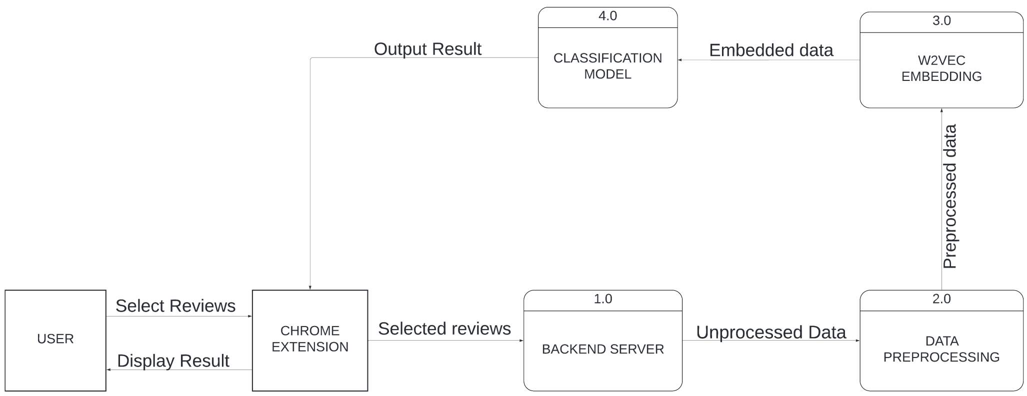

Figure 3.9: DFD Level 1

3.3 Software Development life Cycle

We will be implementing the Incremental Development Model with immediate prototyping. To cope with the future changes, we will develop the application prototype with incremental approach as incremental model is the most widely used model in software development approach.

Incremental Development Model

Incremental model is based on the idea of developing an initial implementation, exposing this to user content and evolving through several versions until a complete system has been developed. It interleaves the activity of specification, development, and validation. It is developed as a series of version(increments) with each version adding functionalities to the previous one.
This model is designed, implemented, and tested incrementally until the product is finished. Any iteration of the increment model can use a prototyping model for the process flow.

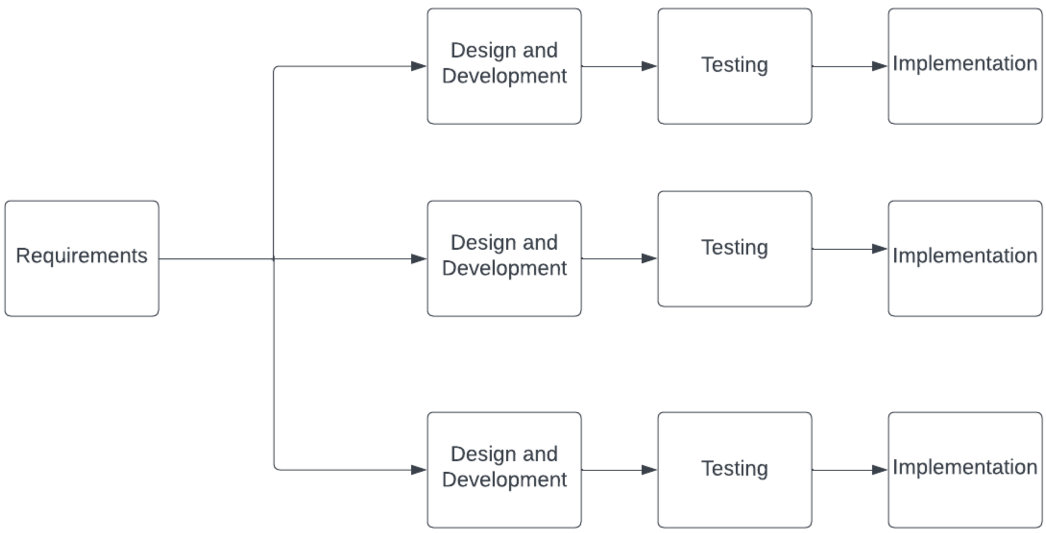

Figure 3.10: Incremental Model

CHAPTER 4
RESULT AND ANALYSIS

4.1 Confusion Matrix

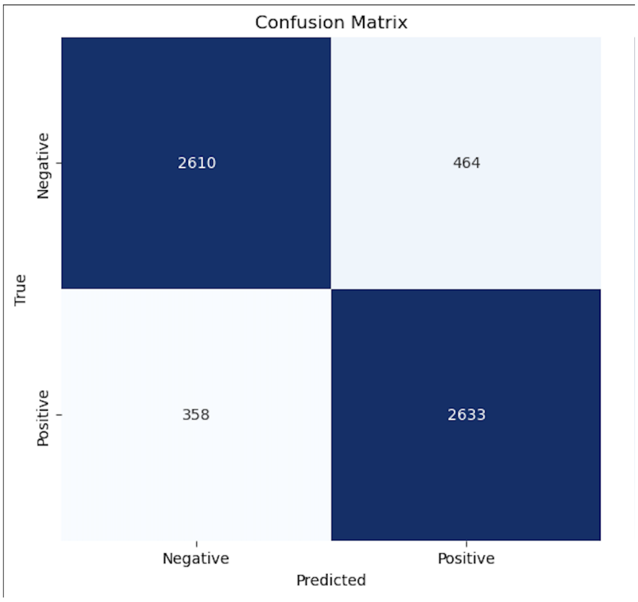

Figure 4.1: Confusion Matrix

4.2 ROC curve

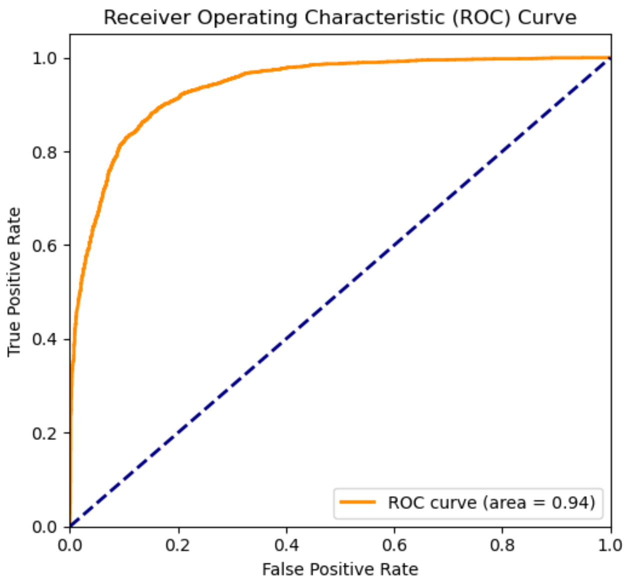

Figure 4.2: ROC Curve

4.3 Training And Testing Analysis

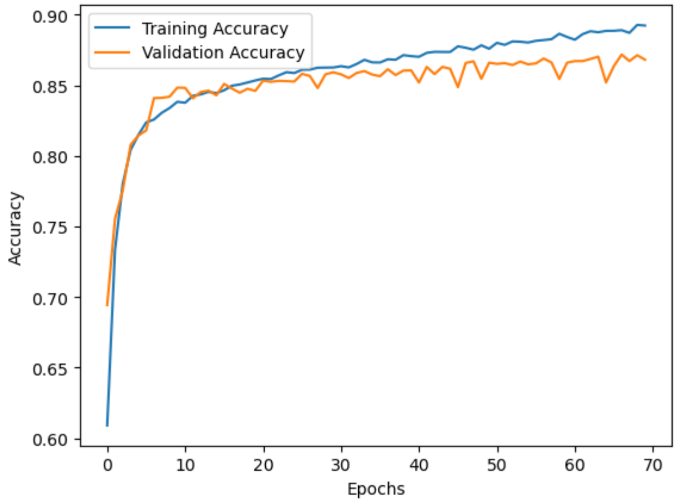

Figure 4.3: Accuracy Chart

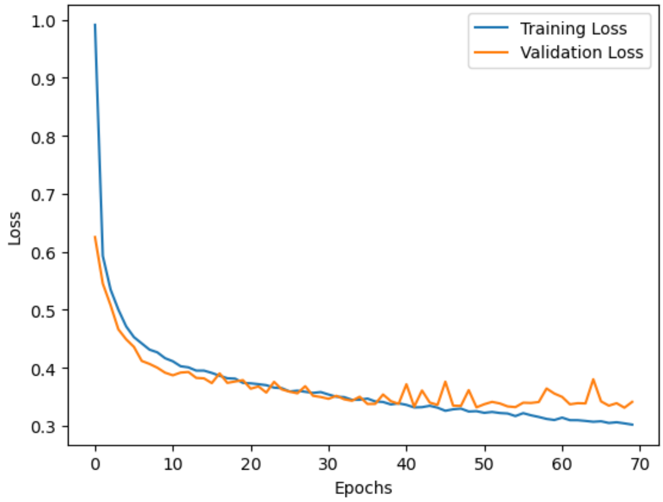

Figure 4.4: Loss Chart

4.4 Evaluation Metrices

F1 Score:0.864

Accuracy: 0.864

Precision: 0.850

Recall: 0.880

CHAPTER 5
CONCLUSION AND FUTURE ENHANCEMENT

5.1 Limitations

Some of the key limitations of our project are:

- Users need to manually select reviews to check their authenticity.
- The system currently only works on one language.
- It demonstrates average performance in real-life data, especially while classifying fake reviews.

5.2 Future Enhancement

We can improve this project in future by using more advanced natural language processing (NLP) models. These models, especially those built on transformer architectures, have shown better performance in various language-related tasks. Another way to boost our project is by improving the quality and variety of our training data. We can do this by adding artificial reviews created by the latest version of GPT and other language model.

5.3 Conclusion

We have developed a product review classifier incorporating a Bi-LSTM Model, as demonstrated through utilization of an Amazon dataset. Our primary objective focuses on the detection of artifically generated reviews within e-comerce platforms, thereby fortifying user confidence. The user-friendly extension facilitates real-time analysis, providing assist to consumers in making well-informed purchasing decisions. Performance metrics, including F1 score, recall, precision, and accuracy, were assessed, yielding an accuracy of 86.4 percentage on unseen test data. Looking ahead, the prospect of refining performance lies in the deployment of advanced deep neural network techniques and more advance algorithms. Our insights are drawn from academic journals and research papers, underscoring the robust foundation of our product review classifier.

REFERENCES

[1] Author(s), "'fake product review detection'," *International Research Journal of Modernization in Engineering Technology and Science*, 2023, preprint.

[2] D. I. Adelani, H. Mai, F. Fang, H. Nguyen, J. Yamagishi, and I. Echizen, "Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection," in *Advances in Knowledge Discovery and Data Mining*, January 2020, pp. 1341–1354.

[3] E. Elmurngi and A. Gherbi, "An empirical study on detecting fake reviews using machine learning techniques," in *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, Aug 2017.

[4] J. Salminen, C. Kandpal, A. M. Kamel, S. Jung, and B. J. Jansen, "Creating and detecting fake reviews of online products," *Journal of Retailing and Consumer Services*, vol. 64, p. 102771, Jan. 2022. [Online]. Available: https://doi.org/10.1016/j.jretconser.2021.102771

[5] A. Al Bataineh, V. Reyes, T. Olukanni, M. Khalaf, A. Vibho, and R. Pedyuk, "Advanced misinformation detection: A bi-lstm model optimized by genetic algorithms," *Electronics*, vol. 12, no. 15, p. 3250, 2023.

[6] R. Haque, S. H. Laskar, K. G. Khushbu, M. J. Hasan, and J. Uddin, "Data-driven solution to identify sentiments from online drug reviews," *Computers*, vol. 12, no. 4, p. 87, 2023.

[7] H. Qayyum, F. Ali, M. Nawaz, and T. Nazir, "Frd-lstm: a novel technique for fake reviews detection using dcwr with the bi-lstm method," *Multimedia Tools and Applications*, Mar. 2023. [Online]. Available: https://doi.org/10.1007/s11042-023-15098-2

APPENDIX

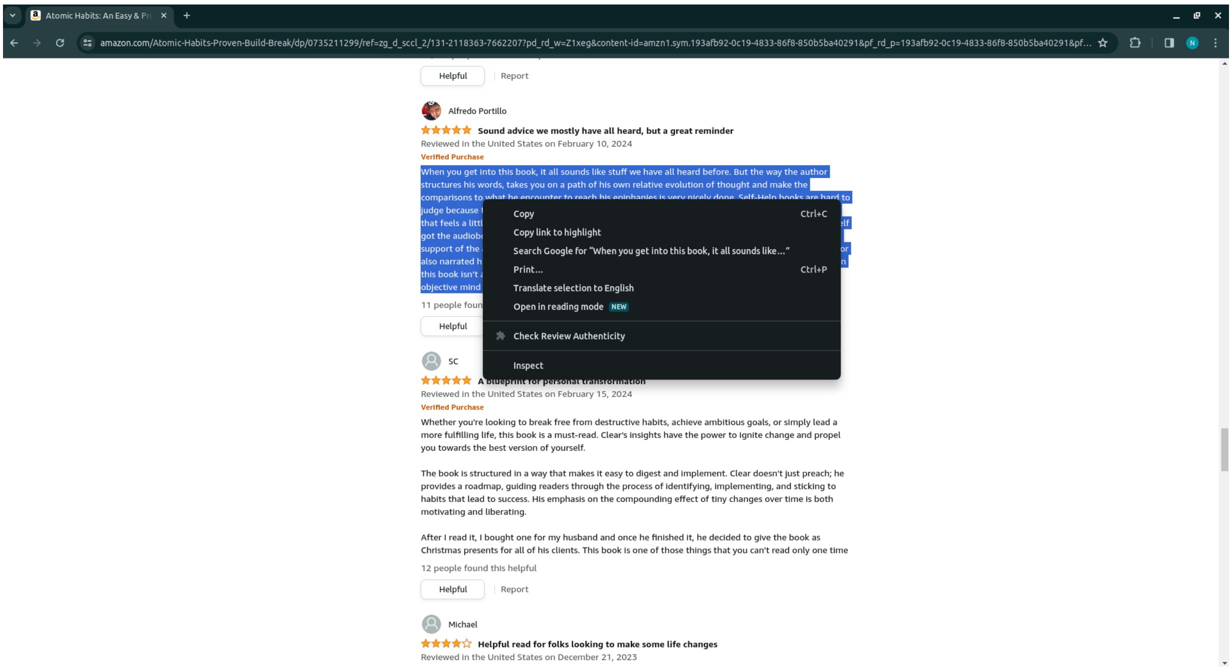

Figure 5.1: Checking Review Authenticity

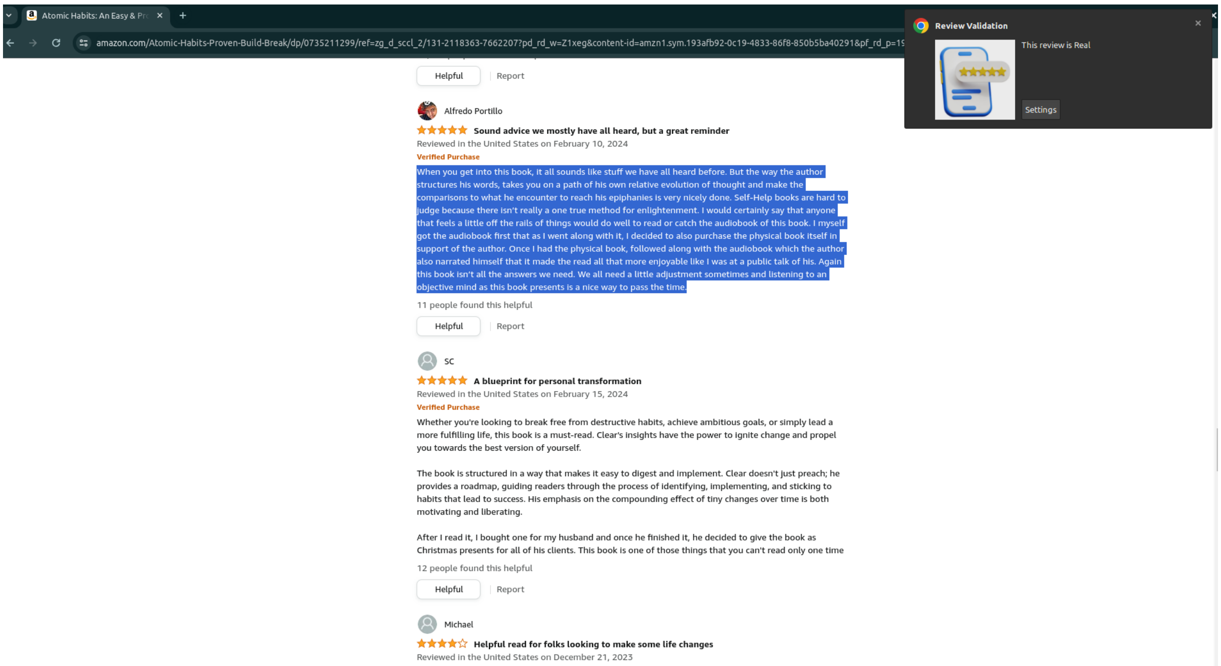

Figure 5.2: Result On Amazon Review

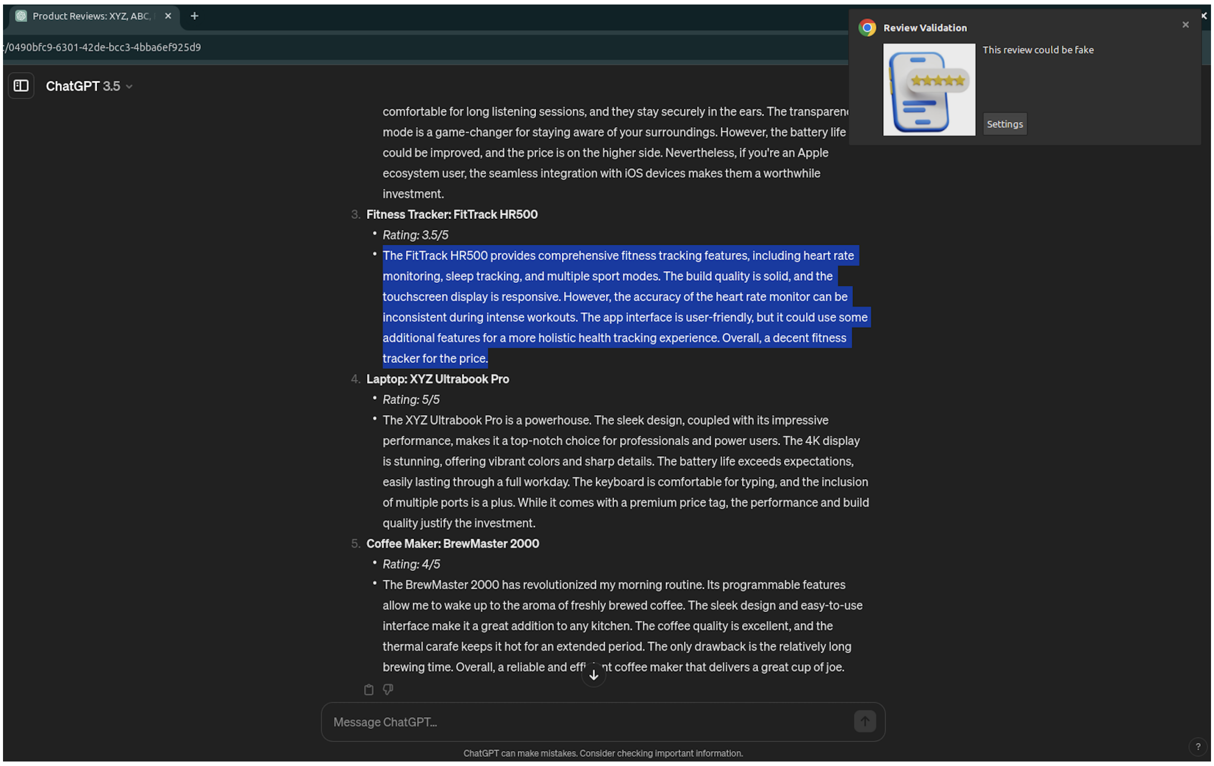

Figure 5.3: Result On GPT Generated Review