TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Report

On

Fake News Detection Using BERT Model

Submitted By:

Abhishek Chaudhary (Exam Roll No. 28152)

Adhip Bhattarai (Exam Roll No. 28153)

Kshitiz Poudel (Exam Roll No. 28167)

Nandan Singh (Exam Roll No. 28169)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

March, 2023

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Report

On

Fake News Detection using BERT Model

Submitted By:

Abhishek Chaudhary	(Exam Roll No. 28152)
Adhip Bhattarai	(Exam Roll No. 28153)
Kshitiz Poudel	(Exam Roll No. 28167)
Nandan Singh	(Exam Roll No. 28169)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

Under the Supervision of

Mr. Kshetraphal Bohara

March, 2023

DECLARATION

We hereby declare that the report of the project entitled **“Fake News Detection using BERT Model”** which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus,** in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Computer Engineering** is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than those listed here have been used in this work.

Abhishek Chaudhary (Exam Roll No. 28152) ______________________

Adhip Bhattarai (Exam Roll No. 28153) ______________________

Kshitiz Poudel (Exam Roll No. 28167) ______________________

Nandan Singh (Exam Roll No. 28169) ______________________

Date: March, 2023

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus,** a minor project work entitled **“Fake News Detection using BERT Model”** submitted by **Abhishek Chaudhary, Adhip Bhattarai, Kshitiz Poudel** and **Nandan Singh** in partial fulfillment for the award of Bachelor’s Degree in Computer Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor’s degree in Computer Engineering.

____

Project Supervisor
Mr. Kshetraphal Bohara
Department of Electronics and Computer Engineering, Thapathali Campus

____

External Examiner
Mrs. Anku Jaisawal
Department of Electronics and Computer Engineering, Pulchowk Campus

____

Project Co-ordinator
Mr. Umesh Kanta Ghimire
Department of Electronics and Computer Engineering, Thapathali Campus

____

Mr. Kiran Chandra Dahal
Head of the Department,
Department of Electronics and Computer Engineering, Thapathali Campus
March, 2023

COPYRIGHT

ACKNOWLEDGEMENT

We would like to express our thanks and gratitude towards the Institute of Engineering, Tribhuvan University for motivating us in doing this project by including a minor project in the course of Bachelor in Computer Engineering.

We would like to express our sincere gratitude to the Department of Electronics and Computer Engineering, Thapathali Campus for providing us with the opportunity to undertake this project. Their guidance and support throughout the project were invaluable and helped us to overcome many challenges. The insightful feedback and suggestions provided by the department's faculty members greatly aided in the successful completion of this project. We are grateful for the valuable guidelines and support that we received from the department and its faculty members, which have contributed to our academic growth and learning.

Our sincere gratitude goes out to Mr. Kshetraphal Bohara, our supervisor, for his continuous support and important advice throughout the project. His knowledge, perceptions, and suggestions played a crucial role in determining the project's course and assuring its success. We were able to overcome several obstacles and go forward with confidence thanks to his willingness to share his assets and offer us quick feedback. We are appreciative of his guidance and encouragement, which have made a big difference in our ability to advance both personally and professionally. We are grateful to have had the chance to work with him because his contributions to this project have been invaluable.

Abhishek Chaudhary (THA076BCT003)

Adhip Bhattarai (THA076BCT004)

Kshitiz Poudel (THA076BCT018)

Nandan Singh (THA076BCT020)

ABSTRACT

Spreading fake news can have different harmful effects. There is a lack of credible platforms for detecting fake news. In this project, we have developed such a system using a combination of source checking and the BERT pre-trained model for the Nepali language and implemented it on the website. Google's BERT model is a state-of-the-art natural language processing tool, pre-trained on vast amounts of text data and applied to both similarity checks and fake news detection. The first procedure involved searching for the provided title in Google, scraping the resulting titles and URLs, and performing a similarity test between the provided title and the scraped titles. If the similarity score exceeded the established threshold value, the news source was examined to determine whether it was a trustworthy source or not. The news was deemed authentic if it came from a reliable source. The BERT model was used to determine whether the news was real or fake if the similarity score was below the threshold value or if the source was doubtful. A pre-processed, tokenized dataset of Nepali text news was used to train the BERT model, which was subsequently optimized through hyperparameter tuning. The BERT model's performance was evaluated using accuracy, precision, recall, and F1 Score metrics. The model achieved an accuracy of 96%, a precision of 97% for real news and 95% for fake news, and the recall was 94% for real news and 98% for fake news, while the f1-score was 96%. Following this, the model was deployed on a website.

Keywords: BERT, Fake, news, Real, online, website

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADAM:	Adaptive Moment Estimation
BBC:	British Broadcasting Corporation
BERT:	Bidirectional Encoder Representations from Transformers
CLS:	Classification
CPU:	Central Processing Unit
CSS:	Cascading Style Sheets
CUDA:	Compute Unified Device Architecture
cu-DNN:	CUDA Deep Neural Network library
GELU:	Gaussian Error Linear Unit
GLUE:	General Language Understanding Evaluation
GPU:	Graphical Processing Unit
GUI:	Graphical User Interface
HTML:	Hypertext Markup Language
LSTM:	Long short-term memory
TPU:	Tensor Processing Unit
NLP:	Natural Language Processing
NLTK:	Natural Language Toolkit
RAM:	Random Access Memory
ReLU:	Rectified Linear Unit
RNN:	Recurrent Neural Network

SEP: Separator

SSD: Solid State Drive

SVM: Supported Vector Machine

UI: User Interface

UNK: Unknown

URL: Uniform Resource Locator

1. INTRODUCTION

The practice of getting news from social media platforms has grown in popularity in recent years. Given the abundance of information at our disposal, it is not surprising that more people use social media for its accessibility and ease than they do traditional news sources. It's crucial to remember, too, that relying only on social media for news can present its own set of difficulties. Fake news can spread faster and easily on social media since there isn't any source checking or credibility there. Particularly when politics and public opinion are at stake, this can result in disinformation and be harmful. Additionally, traditional news sources such as newspapers have established a level of credibility and accuracy through years of rigorous reporting and verification. Although they may take longer to obtain, the reliability and accuracy of traditional news sources cannot be overlooked. Ultimately, it is up to individuals to weigh the convenience of social media news against its potential drawbacks, and to make informed decisions about where they obtain their information.

1.1. Background

As the world becomes more connected through the internet and smart devices, the way we consume news and information is changing. While social media platforms provide an easy and accessible way to stay up-to-date on current events, the reliability of the news circulated through these platforms is often called into question. It's no secret that fake news, propaganda, and misinformation are prevalent on social media, and can have a significant impact on society as a whole. Some websites and social media platforms are specifically designed to spread fake news and manipulate public opinion. These false stories can be incredibly convincing, often presenting a mixture of true and false information in a way that is difficult to distinguish. The consequences of this can be severe, as false information can spread rapidly, leading to confusion and distrust among the general public. Politicians and other influential figures are also known to use fake news to their advantage, either by spreading false information to gain votes or undermining their opponents. The rise of fake news has become such a significant problem that even respected publications such as Forbes magazine have published articles warning readers about the dangers of fake news and providing tips for spotting it. [1] As such, it is more important than ever to remain vigilant when consuming news from social media platforms and to seek out reliable sources of information whenever possible.

Fake news has been around for as long as human beings have been communicating with one another. However, with the rise of social media and the internet, it has become easier than ever to spread false information and manipulate public opinion. One of the reasons that fake news can be so convincing is that it often seeks to divide people into different groups, each of which believes that their opinion is the correct one. This can be particularly damaging in cases where politics or other sensitive topics are involved, as it can lead to a breakdown in communication and a lack of understanding between different groups of people.

Another reason that people often fall victim to fake news is that they are not present to witness the events themselves. Instead, they rely on sources such as news outlets or social media posts to provide them with information about what happened. In some cases, these sources may intentionally or unintentionally provide false information, leading people to believe things that are simply not true. This can be particularly dangerous in situations where people need accurate information to make decisions that will affect their lives or the lives of others.

Finally, it is important to recognize that fake news often targets people's emotions rather than their rational thinking. By playing on people's fears, prejudices, or desires, fake news can be incredibly persuasive, even when it has no basis in reality. This can be particularly dangerous in situations where people are already feeling vulnerable or uncertain, as it can lead to them making decisions based on false information. As such, it is important for individuals to be aware of the potential for fake news and to take steps to verify the information they receive before accepting it as true. There are various types of fake news:

- Clickbait
- Sponsored content
- Fabricated journalism

A typical characteristic of fake news is that it challenges people's belief and try to affect their opinion. They are too good to be true and have grammatical mistakes. They are not published by a reliable source either. They are often the news without facts or misleading or false facts. They are often reported by unreliable people. They may even contain a picture made using editing software to look like a real picture. They are often grammatically incorrect.

1.2. Motivation

During the Mangsir month election in Nepal, false claims and misinformation were spread to gain political influence. Given the critical importance of the election for the maintenance and promotion of democracy, it became increasingly clear that action needed to be taken to prevent the spread of such misinformation. Even online Khabar, a prominent news outlet, published an article addressing the issue of fake news during the election period. [2] The article highlighted the need for responsible journalism and urged readers to be cautious and discerning when consuming news and information related to the election.

The spread of false claims and misinformation during an election has serious implications for the democratic process, as it can undermine the legitimacy of the election and erode public trust in the electoral system. To ensure that the election is fair and transparent, accurate and reliable information must be provided to the public. This requires a concerted effort on the part of journalists, news outlets, and the government to combat the spread of fake news and to promote responsible journalism. By doing so, we can help to ensure that the election is conducted in a manner that is consistent with democratic principles and that the results are accepted as legitimate by all members of the public.

1.3. Problem Definition

The spread of false information can cause harm to people's reputations, result in financial and psychological distress, worsen polarization, and even incite violence. In the context of Nepal, there is a lack of a user-friendly platform to identify such news. This is concerning as the lack of the ability and resources to evaluate news sources critically and identify fake news, leads people susceptible to the spread of false information. This project aims to create a platform that can differentiate between genuine and false Nepali news from given titles and text through Source checking and language processing BERT Model which will enable the platform to classify accurately.

1.4. Project Objectives

The following were the objectives of our project: -

- To distinguish real and fake Nepali news through source checking and BERT model.
- To implement the system on the web and make it widely accessible.

1.5. Scope and Applications of Project

In this age of the internet and social media, fake news spreading is a real issue. And there are not so many fake news identifiers for Nepali news. This fake news detection web application will efficiently detect whether Nepali text-based news is true/false.

For this project, we used a pre-trained BERT model that will further be trained on Nepali text-based real and fake news.

For training our model, we have translated English Fake news to Nepali using Google-translate. so, our model might be inefficient in predicting fake news. Since the Nepali language has its own dialect, it becomes challenging to train the model which will identify such dialects.

The application of this project is as follows:

- Media and Journalism: maintaining the credibility and integrity of journalism through spotting false information in news stories, social media, and other sources.

- Political campaign: identifying and countering political disinformation and propaganda spread during elections and campaigns.

- Business: identifying false information in marketing, advertising, and financial reporting to uphold accountability and transparency.

1.6. Report Organization

In the introduction section, we discussed the trend of obtaining news from social media platforms and the challenges of relying solely on social media for news. It emphasizes the prevalence of fake news, propaganda, and misinformation on social media and the potential consequences of consuming such information. This section highlights the reasons that make fake news convincing, including the division of people into different groups and targeting people's emotions. This section also identifies the characteristics of fake news and highlights the importance of verifying information before accepting it as true.

A literature review is a critical analysis and evaluation of existing research studies, articles, books, and other sources related to a particular research topic. It provides an overview of the current state of knowledge on the topic and helps researchers identify gaps in the existing literature. The literature review also serves as a foundation for the research, providing a context and rationale for the study. By analyzing and synthesizing the findings of previous studies, researchers can identify patterns, inconsistencies, and areas for further investigation. Additionally, a literature review allows researchers to gain insights into the methodologies and research designs used by others in the field, which can inform their research approach.

The requirement analysis section of a project is an essential component that provides critical information about the feasibility of the project on multiple fronts, including technical, economic, and operational. This section serves as a foundation for the development of a successful project by outlining the objectives and goals of the project, assessing the available resources, and identifying the constraints and challenges that may arise during the implementation process. The hardware and software requirements of the project are also identified in this section, ensuring that the project team has a clear understanding of the tools and technologies needed to meet project objectives. By conducting a comprehensive requirement analysis, we have ensured that our project is well-planned, well-structured, and well-resourced, setting the stage for a successful outcome.

The section titled "System Architecture and Methodology" presents the approach utilized for the development of our project, including a block diagram, flowchart, algorithms, and the model employed. The methodology involved initially searching the title provided by the user in Google and comparing the similarity between the user's title and those found on Google. If the similarity exceeded a pre-determined threshold and the news originated from a trusted source, it was presented as authentic. Alternatively, the BERT model was utilized to classify the news as real or fake if it did not meet these criteria.

In implementation details, we provided a comprehensive and detailed description of how the project was designed and executed, and how the different components of the project were integrated to achieve the desired outcome.

The presentation and interpretation of the data and conclusions produced by the project are normally the main goals of the results and analysis part of a project report. To aid readers in understanding the findings, this section frequently includes tables, graphs, and charts.

Future enhancements focus on further improvements or iterations of the project. These enhancements for future work are based on insights gained during the project's implementation and analysis, and they help to identify areas where the project could be improved or expanded.

The conclusion provides a summary of the project's key findings, outcomes, and contributions.

Appendices are typically used to provide additional detail or supplementary information that is not included in the main text of the document, but which may be of interest or relevance to certain readers.

References include detail about sources and materials that were used or consulted during the research and development phases.

2. LITERATURE REVIEW

Random Forest is an ensemble learning method for classification and regression that was introduced by Leo Breiman and Adele Cutler in their 2001 research paper "Random Forests" [3]. The algorithm works by constructing multiple decision trees during the training phase, where each tree is built on a random subset of the input features. The output of each decision tree is the mode of the classes for classification tasks or the mean prediction for regression tasks. During the prediction phase, the Random Forest algorithm takes the input data and applies it to each decision tree in the forest. The individual outputs of each decision tree are then combined to produce the final prediction. The key advantage of the Random Forest algorithm is its ability to reduce the risk of overfitting by reducing the variance and increasing the accuracy of the final prediction. Additionally, the Random Forest algorithm can handle missing values and maintain accuracy even when dealing with high-dimensional data. Its ability to capture complex interactions between variables, while remaining relatively insensitive to noise and outliers, makes it a popular algorithm for a wide range of machine-learning tasks. The work of Breiman and Cutler has laid the foundation for the development and widespread use of the Random Forest algorithm in various fields such as finance, healthcare, and natural language processing.

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that can process sequential data and make use of long-term dependencies. It was specifically designed to overcome the vanishing and exploding gradient problems that can occur when training traditional RNNs, and it has been successful in a wide range of applications including language translation, image captioning, and speech recognition. An LSTM network is composed of "cells" that contain information from the past and pass it on to the present and future. These cells are controlled by three "gates" that decide what information to store, what information to throw away, and what information to pass on. This allows the LSTM to selectively remember and forget information, making it well-suited for tasks that require the retention of long-term dependencies.

BERT [4]stands for Bidirectional Encoder Representations from Transformers is a transformer-based architecture for natural language processing tasks such as language translation, question answering, and language modeling. It was developed by researchers at Google and has been shown to outperform many previous models on a wide range of tasks. The key innovation of BERT is that it is a "bidirectional" model, meaning that it takes into account the context both to the left and the right of a given word. This is in contrast to previous models, which typically only considered the context to the left of a word. This bidirectional approach allows BERT to better understand the meaning of a word in the context of the entire sentence, rather than just based on the words that come before it. BERT is trained on large amounts of data and can learn the structure of the language in an unsupervised manner.

2.1. Previous works

In research performed by Kelly Stahl, it uses Network Analysis, Linguistic Cues, Fact-checking, Naïve Bayes Classifier, SVM, and Semantic Analysis. They used both Naïve Bayes Classifier and SVM. They found the accuracy of the combined methods was more than the accuracy of the individual methods. They found that the biggest drawback of the Naïve Bayes classifier was that it deems all features of a document, or whichever textual format is used, to be independent even though most of the time that is not the situation [5].

Jiawei et al. Conducted a study based on actual textual information, authorship, and article-subject relationship. They introduced the FAKE DETECTOR framework which combines representation feature learning and credibility label interface to compose a deep diffusive network model. They were able to achieve an accuracy of 0.63 [6].

Also, Gowthami. K et. al used SVM and Random Forest to identify fake news. At first, they collected the dataset and then pre-processed the data. They applied SVM and Random Forest algorithm and they compared the results and accuracy [7].

In research conducted by Yesugade, et al. they collected scrap data and transformed the data into the format. With the help of the NLTK toolkit, they removed the stop words. Then they performed word embedding and the word index of the tokenized dataset was generated. In addition to this, they compared RNN-LSTM and sigmoid validation. Here, they used 100 neurons in each layer with each layer using the sigmoid activation function. They classified the news as fake news – 0, and real news – 1 [8].

Uma et al. used combinations of different algorithms like static search, dynamic search, and URL search in the classification of fake News. The static search included machine learning algorithms Like Naive Bayes, Random Forest, and Logistic Regression. Dynamic search asked the users to enter specific keywords in a news and produced a percentage probability of truthfulness of an article by comparing articles with similar keywords on the internet.URL search method accepts a specific website domain and calculates the authenticity of the website compared to the databases of the websites like LIAR, BuzzFeed, and BS Detector. They were able to get up to 80 percent accuracy with the Logistic regression model and 92.73 percent accuracy with passive aggressive classifier [9].

In research conducted by Sastrawan et. al, about the Detection of fake news using deep learning CNN–RNN-based methods they conducted the research in 2 phases i.e., the training phase and the testing phase. The first phase begins by retrieving the training data from the database. The data cleansing is performed on the data. Then they performed a data augmentation process to the cleaned data to balance the data between the classes. This augmented data is then processed and transformed into word vectors. These word vectors are used to train the deep learning model. This model is stored in the database for the next phase. In the second phase, the trained model is evaluated. At first testing, data is pre-processed and the previously stored model is taken from a database. Now, the pre-processed data and saved model are used to predict the pre-processed test data and the results are displayed [10].

In research conducted by Farokhian et. al, at first, they extracted the headlines of the news and fed it to the BERT network. On the other hand, using the Max-Worth algorithm, the most appropriate news text span is selected from the rest of the news text. This text span is again fed to another BERT network. The output from both BERT networks was joined together and fed to the dropout layer. Then the output from the dropout layer was fed to the dense layer which classified the news as real or fake [11].

In research conducted by Dam et. al, they detected click-bait in the Nepali language using SVM and random forest algorithm. At first, they took the news title and news body from the dataset and performed pre-processing on the dataset. They made a collection of unique words from corresponding news titles and news-body called vocabulary. They used TDF-IDF to represent the words in the form of vectors. They calculated the cosine angle between two vectors to measure how similar they were. They used SVM and random forest algorithms to detect clickbait. They obtained an accuracy of 95.03% using SVM and 94.93% using the Random Forest algorithm [12].

Neha et. al, researched fake news detection using various models, namely LSTM, LSTM with Attention Mechanism, BERT, and BERT with LSTM and Attention. They started by pre-processing the data and then passing it to the embedding layer for the LSTM model. The resulting vector from the embedding layer was then input into the LSTM layer, and finally, a dense layer with a softmax activation function provided the desired output. For the LSTM model with an attention mechanism, an attention layer was added before the final dense layer for classification. The researchers also utilized the bare Bert Model transformer, which outputs raw hidden states without any specific heads on top, in their study. The raw output from the Bert Model was passed to a dropout layer, and then a linear layer with a softmax activation function was used to provide the required output. In addition, for the BERT model and LSTM with an attention mechanism, the output of the BERT model was connected to an LSTM layer, followed by an attention layer, and finally, a linear layer was utilized for classification. The results showed that the BERT model had the highest accuracy, precision, recall, and F1 score with a score of 0.895. Following this, the hybrid model, which was the BERT with LSTM and Attention model, performed well with precision, recall, and F1-score of 0.820, 0.821, and 0.818 respectively [13].

3. REQUIREMENT ANALYSIS

3.1. Project Requirements

3.1.1. Hardware Requirements

In order to build a BERT for fake news detection, it is essential to consider the hardware requirements necessary for efficient data processing, model training, and evaluation. Several factors such as the size of the dataset, model complexity, and several training and inference iterations can affect the hardware requirements for this task.

- To ensure efficient data pre-processing, model training, and evaluation, a powerful CPU with multiple cores is highly recommended. The use of a high-end CPU can significantly reduce the processing time required for these tasks, thereby enhancing the overall efficiency of the process.
- In addition to a powerful CPU, a GPU is highly recommended for training and inference, as BERT models are computationally intensive. A CUDA-enabled GPU, such as NVIDIA with at least 4GB of memory, is recommended for this purpose. By using a GPU, users can significantly reduce the training and inference time required for the model, thus achieving faster results.
- Furthermore, memory is also an important consideration when using BERT for fake news detection. It is recommended to have a minimum of 8GB of RAM to ensure smooth data processing, model training, and evaluation. Depending on the size of the dataset, more memory may be required to ensure optimal performance.
- In terms of storage, a high-capacity storage device such as an SSD is recommended to store the dataset, pre-trained models, and trained models. This ensures easy access to the required data and models, thereby improving the overall efficiency of the process.
- Finally, a high-speed internet connection may be necessary for downloading the pre-trained BERT models and updating the libraries and packages required for utilizing them. This ensures that the most up-to-date and accurate models are being used for the task at hand, which is essential for achieving accurate results.

3.1.2. Software Requirements

The software requirements for using BERT for fake news detection in Python include:

1. Python: Python is a well-liked programming language with a sizable community and includes a ton of NLP and machine learning packages, making BERT implementation simpler.

2. TensorFlow: For certain pre-processing and assessment activities, TensorFlow is necessary.

3. Pandas: Pandas are used to clean and manipulate data.

4. Matplotlib, Seaborn, or other visualization libraries: These libraries can be used to create visualizations of the dataset and the model's performance.

5. CUDA and cu-DNN: CUDA and cu-DNN must be installed to use a GPU. For generic processing on GPUs, NVIDIA has created the parallel computing platform and programming language known as CUDA. Deep neural network primitives are available in the cu-DNN GPU-accelerated library.

6. HTML, CSS, Javascript, Flask: To construct a website we would use this software.

3.2. Feasibility Analysis

1. Data availability:

For our BERT model which helps to understand the context of the given text, we scraped real news data from various sources like e-Kantipur, Ghorkhapatra, Nagarik, etc., and for fake news, we translated the news from Kaggle in the English language into Nepali language using Google translate. Besides this, we had a few fake news datasets scraped from clickbait sites. For source checking the user provides the data

2. Financial Feasibility:

All the libraries we need for the completion of this project are open-source. So, there is no significant cost for the completion of our project. Also, the hardware requirements are minimal, so we do not have to buy any additional hardware equipment for the completion of our project.

3. Technical Feasibility:

Currently, we have a laptop with the following specifications:

i. Processor: 11th Gen Intel® Core™ i7-11800H @ 2.30 GHz(16 CPUs) ~ 2.3 GHz

ii. RAM: 16 GB

iii. Storage: 512 GB SSD

iv. Graphics: NVIDIA GEFORCE RTX 3050 Ti (4 GB memory size)

v. Network: We have a fast and stable internet connection on our laptops.

For software requirements, we have installed all the necessary packages required for the completion of our project.

4. Operational Feasibility:

Since the project will have an easy-to-use GUI, any person without technical knowledge related to machine learning can access it. Our system will predict the truthfulness of news with a machine learning model and using source checking, but all the details will be hidden from the end user. The end user will just have to provide the news title and text. Therefore, almost anyone can operate the system. So, we can say that our system is operationally feasible.

5. Schedule Feasibility

The project can be completed within the allocated period for the minor project which is around 3-4 months as per the timelines shown in the Gantt chart.

4. SYSTEM ARCHITECTURE AND METHODOLOGY

4.1. Block Diagram

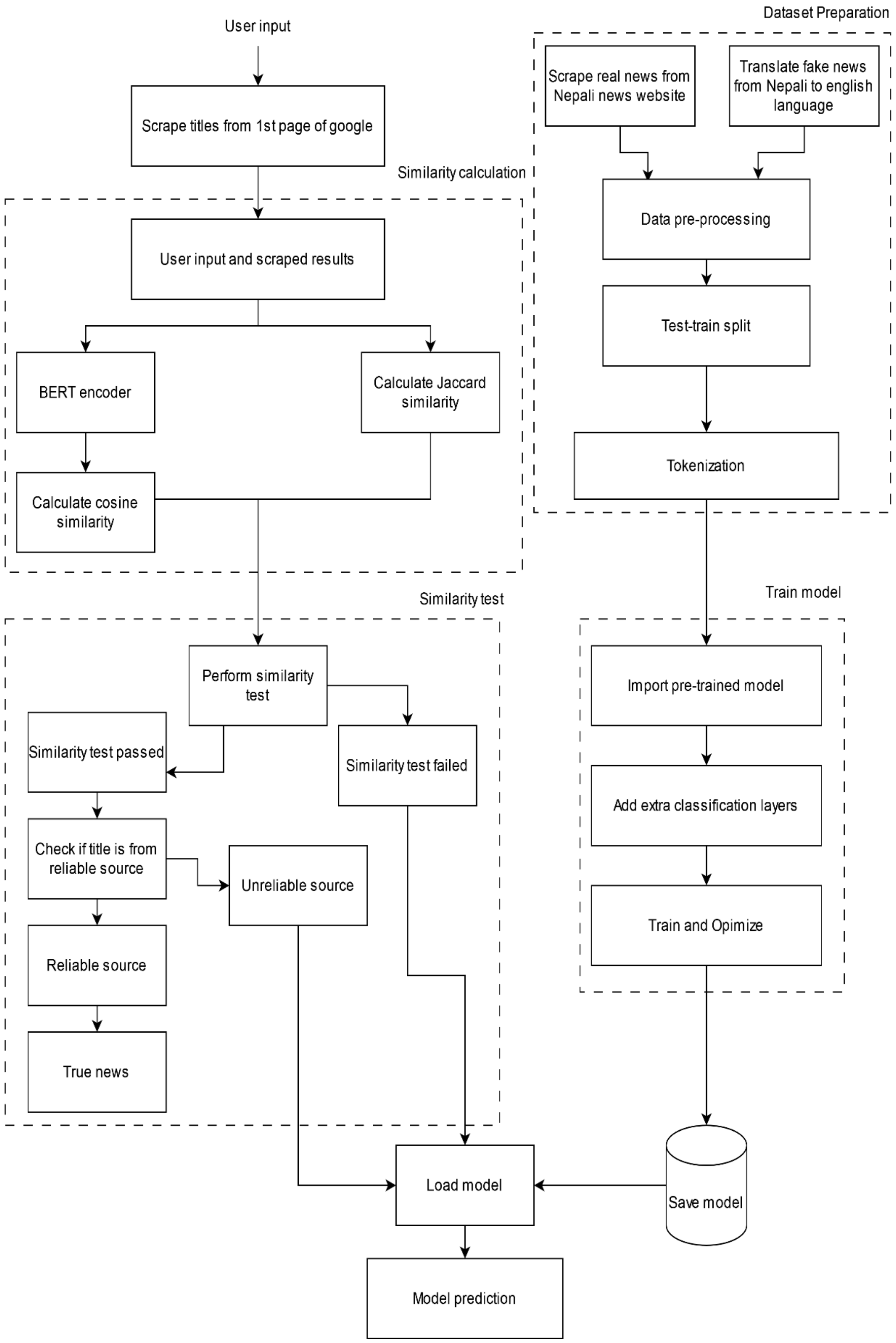


Figure 4-1. Proposed methodology

Our system classifies news based mainly on the context of the news and by checking similar news on other sources on the web. This will ensure that our prediction will be very reliable. For a semantic understanding of news articles, we have used the BERT multilingual pre-trained model. BERT (Bidirectional Encoder Representations from Transformers) as its name suggests is an architecture using only the encoder part of transformers. BERT consists of a stack of encoder layers. where each encoder layer consists of multi-head attention and a feed-forward neural network.

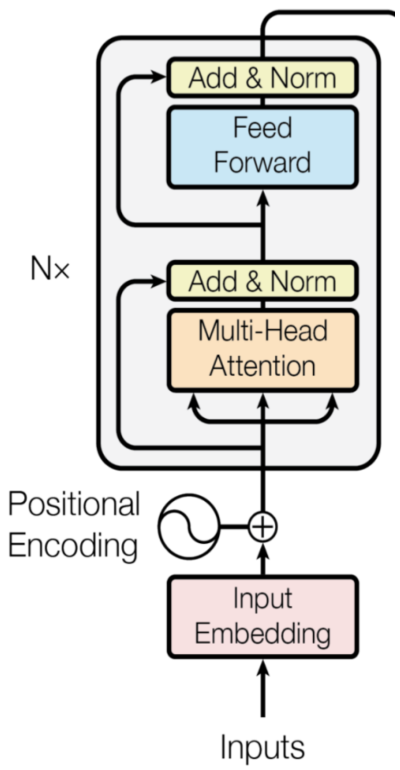


Figure 4-2. Encoder architecture

BERT has been proven to be very good for NLP tasks because it allows truly bidirectional context understanding. Unlike other models which process sequence from left to right only or combine the representation of left-to-right and right-to-left, BERT processes the whole input at once. This allows BERT to capture the meaning and context of words in a sentence more effectively. On the other hand, the whole input can be passed at once instead of sequence to sequence. This helps in better utilization of GPU which is designed for parallel processing. Also, BERT can handle long-range dependencies and attend to specific parts of the input, which are important for understanding the context of a statement and determining whether it is true or false.

The training of BERT is for understanding the semantics between input text and predicted output i.e., real or fake. Since the BERT model is already pre-trained in the Nepali language, we can say that it understands the relationship between different Nepali words, and how the combination of them can generate a certain meaning. The main purpose of training the architecture on real and fake news datasets is to let the model understand certain features of fake news and real news. With the semantic understanding of the Nepali language, the BERT model will generate vectors for given news text. Our model aims to capture the underlying patterns between these vectors of real and fake news that will distinguish them.

Only the context or semantics of a news article is not enough to fully classify the news. For example, some news about events may be true today but not after a particular amount of time. If a piece of news is well crafted the semantics may not be very different. To overcome this problem, we will compare the given input news article with similar news articles scraped from different Nepali news sources. The algorithm for this method looks like this:

Step 1: Input the news title and description from the user.

Step 2: With the given title, google search for similar news/articles.

Step 3: Scrape the news title and URLs obtained from step 2.

Step 4: Compute the similarity between the input news title and the list of news titles obtained from step 2, using cosine similarity and Jaccard similarity.

Step 5: Create a set ‘S’, that contains the trusted Nepali newspaper URL

Step 6: Classify the news as true and Fake.

> Step 6.1: True if the similarity is greater than the threshold value and the news is from a trusted newspaper
>
> Step 6.2: If 6.1 fails, then run the BERT model on the description of the input and check if the news is true based on the context

4.2. Description of the Working Principle:

This Fake news detection has two features to detect news as true/fake. Firstly, we use source checking of the news by web scrapping method.

4.2.1. Web Scrapping Method:

This method is straightforward and it doesn't require any training or testing data. Input news titles and descriptions are taken from the user. With the given title, we import Google search in python and scrape all the related news articles. Then we perform similarity checking for the user-given title and title obtained from a search using the BERT model. Cosine similarity and Jaccard similarity are used for similarity checking.

4.2.1.1. Cosine similarity:

The metric employed in our system for assessing similarity between text documents involves computing a value within the range of -1 to 1 for BERT embedding vectors. The use of the tf-idf approach would result in a value of 0-1 instead. In particular, a value of 1 indicates that the vectors are precisely similar, while a value of -1 implies that they are completely dissimilar. As the value of the metric increases, the degree of similarity between the vectors becomes progressively greater. Therefore, the larger the metric value, the higher the degree of similarity between the text documents under consideration. This metric is an important tool in our system for accurately assessing the similarity between news articles.

$$similarity = cos(\theta) = \frac{A.B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=1}^{n} B_i^2}} \ldots\ldots\ldots\ldots\ldots (1)$$

Where,

A= Embedding vector generated from text1

B= Embedding vector generated from text2

4.2.1.2. Jaccard Similarity:

The Jaccard similarity checks if two texts are similar by directly comparing the words. So, it doesn't need the text to be converted into vector form. In literature, Jaccard similarity, symbolized by, can also be referred to as Jaccard Index, Jaccard Coefficient, Jaccard Dissimilarity, and Jaccard Distance. The value ranges from 0 to 1, 0 meaning no similarity and 1 meaning exactly the same words used. Here the Order of words does not matter.

$$Similarity = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \text{................................(2)}$$

Where,

J=Jaccard distance

A= Nepali text 1

B= Nepali text

4.2.1.3. Similarity test:

Cosine similarity measures if two texts are semantically similar or not, while Jaccard similarity measures if used words are similar or not. If the cosine similarity is greater than 0.7 and the Jaccard similarity was greater than 0.5 then we can say that the two news are the same. After checking the similarity of the news titles, we check if the news is from reliable sources such as eKantipur, gorkhapatraonline, Setopati, etc. If the news is similar and from reliable sources then we called as true. Otherwise, we implement our BERT model to test the given text as real or fake.

4.2.2. BERT Model:

On occasion, news that is not available on search engines like Google may be found on social media. Such information can come from regional news organizations that are less trustworthy than eKantipur, Setopati, or Ratopati. Yet these reports may be accurate. We may utilize the BERT model, a sophisticated language model that examines the context of the news to evaluate if it is real or fake, to confirm the veracity of such material. Even if the news source is not reliable, this technology can assist us in discerning fact from fiction.

The working detail of training of Bert model with extra classification dense and dropout layers to classify news based on the semantics of the Nepali language is described below:

4.2.2.1. Data collection:

We scraped Nepali true news from various reliable sources such as ekantipur, onlinekhabar, gorkhapatraonline, etc. Nepali Fake News is not available so we translated English fake news to Nepali using google translate.

4.2.2.2. Data pre-processing:

We labeled the dataset as Real-0 and fake-1. Since the raw Nepali text is not suitable as input directly to the model. we have to modify or preprocess it.

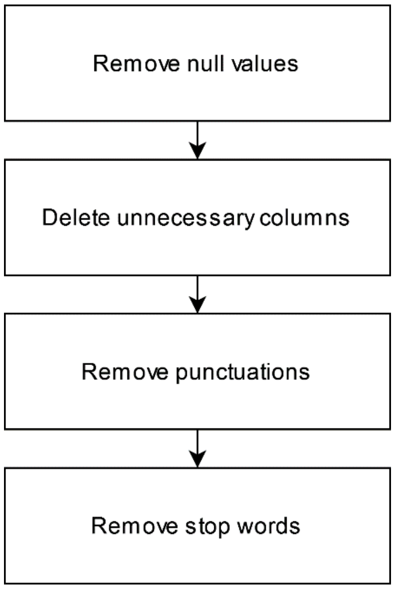


Figure 4-3. Data pre-processing

For pre-processing, we performed the following steps:

- At first, we checked if any value is null in the dataset. If the null value was present, we removed the entire row containing it from the dataset.

- Then we deleted other columns like date, author, etc. from the dataset as they are not relevant to train our model.

- After this we removed punctuations like ':', '?', '&', etc. from our dataset. We performed this process because we have a limited number of tokens available for giving input to the model. So, we need to remove those characters from the text that does not have a significant meaning which will efficiently use the available number of tokens.

- In addition to this, we also performed the removal of stop words like 'उनले', 'का', 'छ', etc. from our dataset. This process is also performed for the same reason as punctuation removal.

- Since the same number of tokens is required regardless of stemming the data or not, we decided not to perform stemming of data. Furthermore, due to the stemming of the data, the context/ meaning of the data can be lost which decreases the accuracy of our model.

4.2.2.3. Test-train split:

Here, we split the dataset into train data and test data in the ratio 8:2. Then again split the training data again into training data and validating data.

4.2.2.4. Tokenization:

First, we convert all the words into their corresponding tokens using the Bert preprocess. Every statement starts with CLS Token and ends with a SEP token. All unknown tokens are replaced by UNK tokens.

4.2.2.5. Define model architecture:

Here, we defined our model architecture which consists of BERT and additional classification layers.

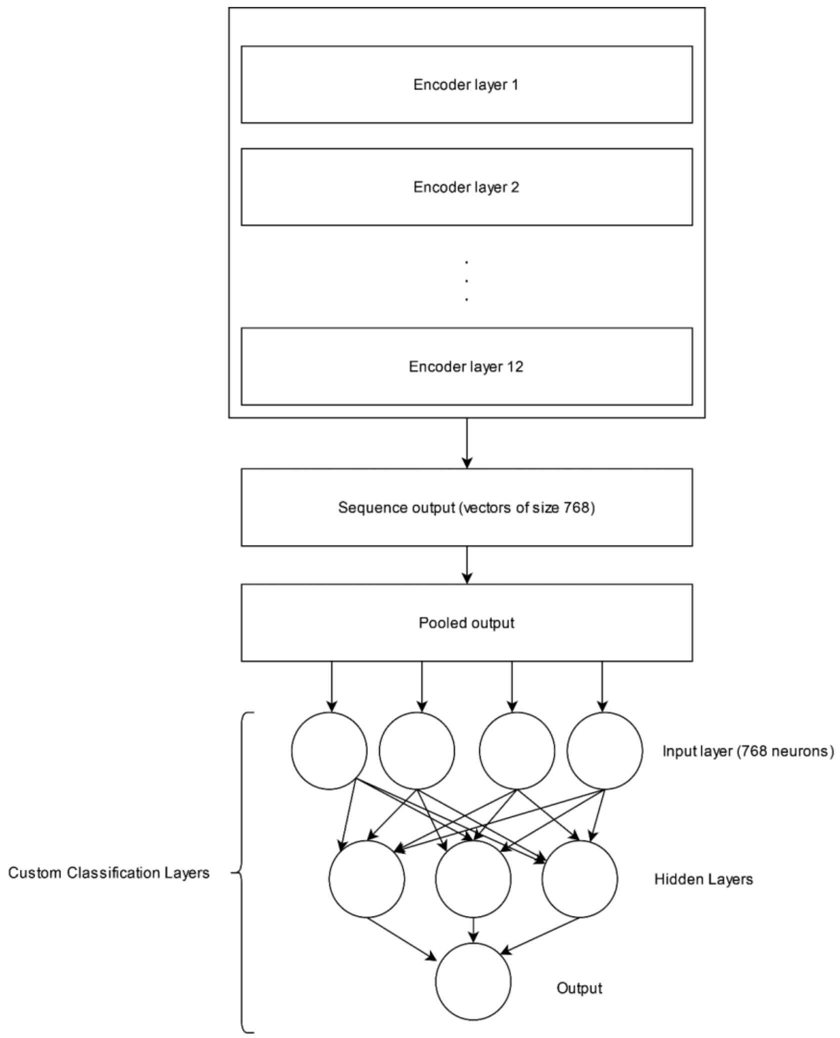


Figure 4-4. Model Architecture

4.2.2.6. Embedding layer:

In the pre-trained BERT multilingual model, the data is passed through the embedding layer. It converts each word in the input to a vector known as ‘word embedding’ that captures the meaning and context of the word. These word embeddings are combined to form a fixed-length vector representation called ‘sentence embedding’ that captures the meaning and context of the sentence. These embeddings are passed to an encoder that understands the meaning and context of the text. The encoder is a stack of transformer layers that use a self-attention mechanism to weigh the importance of each word in the input text and a feed-forward neural network to extract complex features from the input text. The self-attention mechanism allows the model to understand the relationships between words and their context, which is important for accurately detecting fake news. The feed-forward neural network then processes the representation created by the self-attention mechanism, allowing the model to extract more complex features from the input text. Then we added dense and dropout classification layers to the BERT model. We also performed L2 regularization in those added layers so that the model can generalize to a wide range of inputs and prevent over-fitting.

The input to the neural network for classification is the output vector from the BERT. BERT outputs 768-sized vectors corresponding to the number of tokens. We only need the pooled output which is the final vector from the last encoder layer. Therefore, there are 768 neurons in the input layer. The hidden layers are added with dropouts in some layers.

4.2.2.7. Dropout and L2 regularization

Dropout regularization is performed by randomly dropping 10% of the neurons in the hidden layers. This technique helps to prevent over-fitting because the final output cannot be heavily affected by any single neuron to make every prediction close to the actual output.

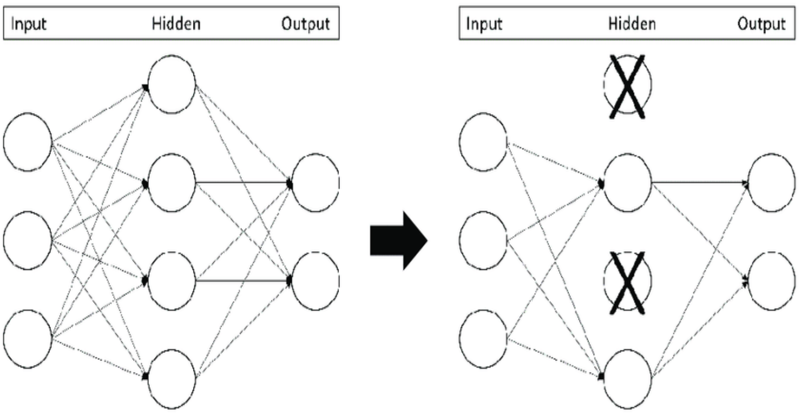


Figure 4-5. Dropout Regulation

In machine learning, L2 regularization, sometimes referred to as weight decay, adds a penalty term to the loss function of a model to prevent overfitting. The penalty term, also known as the "L2 norm" of the weights, is proportional to the square of the size of the model's weights.

The L2 regularization term can be written mathematically as:

$$L2_{reg} = \lambda \times ||\mathrm{w}||^2 \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. (3)$$

The loss function is changed to incorporate the L2 regularization component when training a model with L2 regularization. The model's objective then shifts to keeping the weights as little as feasible in addition to minimizing the initial loss function. Making the model less complicated, helps in preventing the model from overfitting to the training data.

4.2.2.8. Activation Function:

Activation functions introduce non-linearity to the model. This allows the model to learn and represent more complex relationships between the inputs and outputs.

A neural network would be nothing more than a straightforward linear regression model without the activation function. Only linear connections between inputs and outcomes could be learned if there was no activation function. Activation functions also enable a neural network to describe nonlinear and complicated interactions between inputs and outputs.

Additionally, activation functions also help to normalize the output of a neuron, ensuring that it falls within a specific range, such as between 0 and 1 or -1 and 1. This normalization can help to prevent the output of a neuron from becoming too large, which can lead to numerical instability and slow down the learning process.

Therefore, activation functions are essential components of artificial neural networks, and their proper selection and implementation are crucial to the success of the network in solving the problem at hand.

The activation functions used are:

- Sigmoid activation function: This activation function is used in the output layer for binary classification. It gives a value between 0 and 1.
  Its mathematical form is:

$$y = \frac{1}{1 + e^{-x}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

Where x denotes the weighted sum coming to the input of the neuron. Here, output values closer to 0 suggests a more negative input, and values closer to 1 indicate a more positive input.

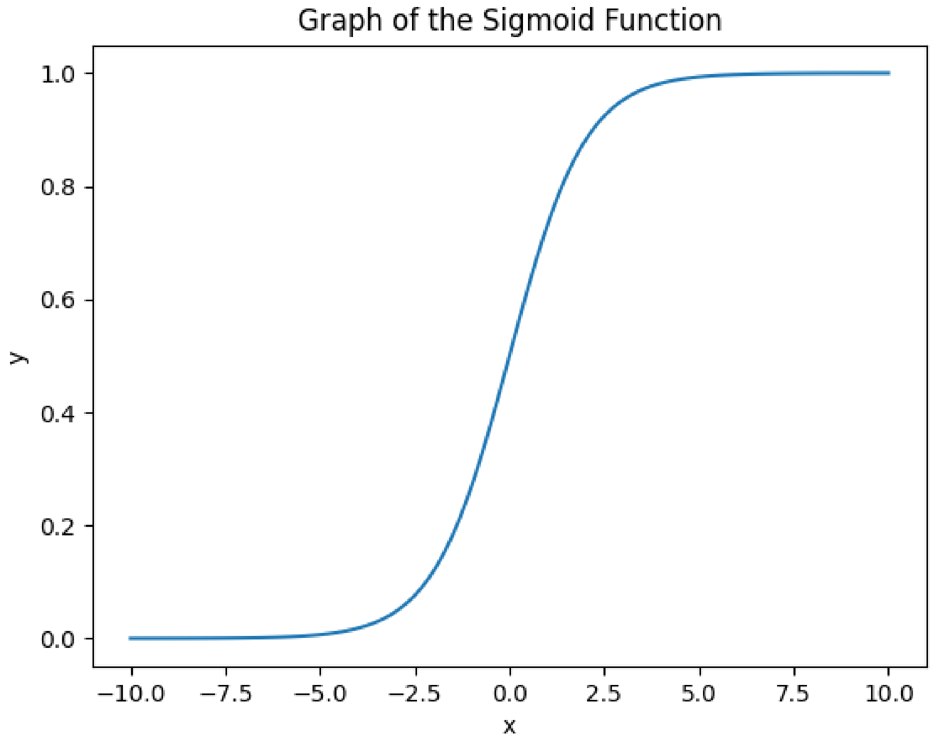


Figure 4-6. Sigmoid Function Graph

- ReLU activation function: This is used in the activation of neurons in the hidden layers of the classification network. Its output and input are the same if the input is positive otherwise 0.

$$y = \max(0, x) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . \dots \dots \dots (5)$$

From the equation, we can see that the ReLU activation function is linear for positive values of x which makes it easy to optimize using gradient-based methods.

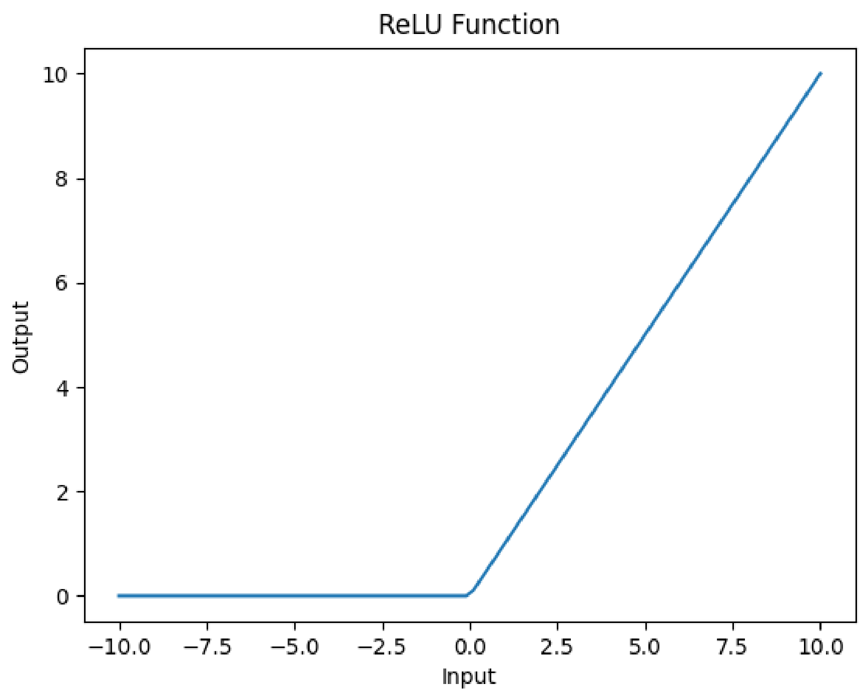


Figure 4-7. ReLU function Graph

- GELU activation function: The pre-trained BERT uses this activation. It is a smooth approximation of the ReLU function. It assumes x to be a normally distributed variable. The output is obtained by multiplying the input and CDF(x) where CDF is the cumulative distributive function that measures the probability that inputs can be less than the given input. In programming, it is often implemented with the error function erf(x).

  Its mathematical form is:

$$GELU(x) = xP(X \leq x) = x\emptyset(x) = x.\frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right] \ldots (6)$$

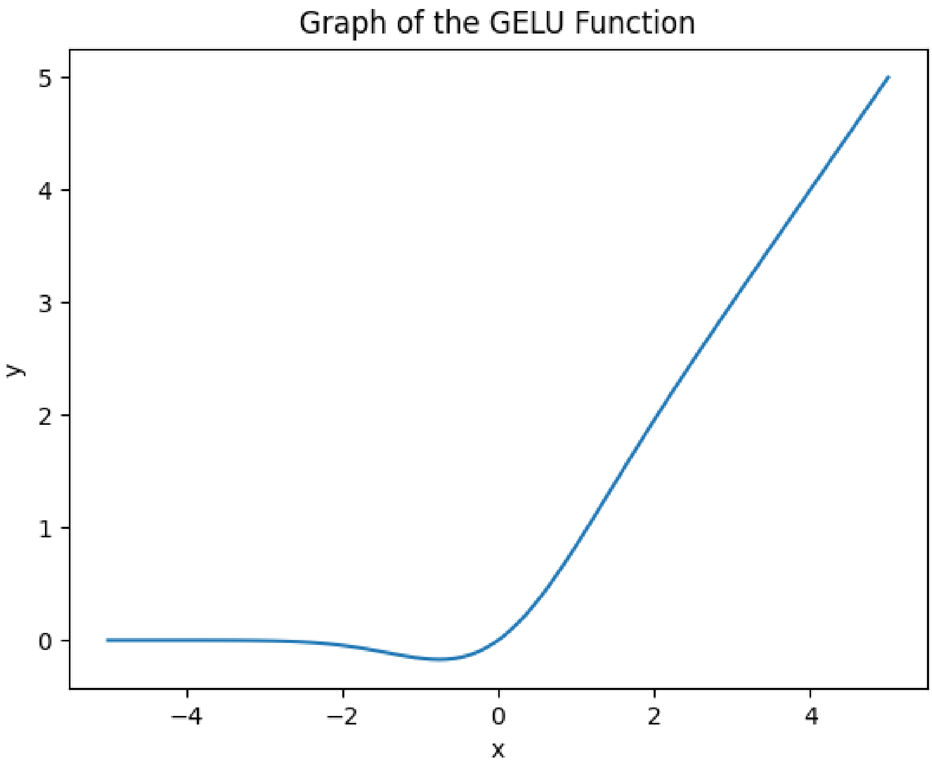


Figure 4-8. GELU Function Graph

4.2.2.9. Loss function:

A loss function is fundamentally very straightforward: It is designed to measure how faulty is the model's output and actual output. The loss function value will be a greater number of inaccurate predictions while low for more accurate predictions. The loss function will indicate whether we are making progress when we modify various aspects of our program to try to enhance our model.

We have used binary cross-entropy to calculate the cost as we only have two classes. The cost function is given as:

$$H_p(q) = -\frac{1}{N}[y_i \, . log\big(p(y_i)\big) + (1 - y_i) \, . \; log \; (1 - p(y_i))] \ldots (7)$$

Here, y is the actual output, and p(y) is the probability of y predicted by our system. Since the fake news class is labeled as 1 and real news is labeled as 0, if the actual model output is 1, the second term will be canceled and cost calculated only by the first term i.e., $y_i \,. log\left(p(y_i)\right)$ which becomes $log\left(p(y_i)\right)$ since y_i is 1. Now, if the model's prediction is closer to 0, the term $log\left(p(y_i)\right)$becomes very high which signifies that the closer the model's prediction to actual output lesser the cost. It works similarly if the actual output is 0.

4.2.2.10. Optimizer:

Optimizer is needed to find the combination of learnable weights and biases that results in the minimum value of the loss function. We have used the ADAM optimizer which computes individual learning rates for each parameter.

The optimization function is given as:

$$w_{t+1} = w_t - \alpha_t m_t \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots..\dots\dots (8)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\left[\frac{\delta L}{\delta w_t}\right] \dots\dots\dots\dots\dots\dots..\dots\dots\dots\dots. (9)$$

Where,

m_t = aggregate of gradients at time t [current] (initially, m_t = 0)
m_{t-1} = aggregate of gradients at time t-1 [previous]
W_t = weights at time t
W_{t+1} = weights at time t+1
α_t = learning rate at time t
∂L = derivative of Loss Function
∂W_t = derivative of weights at time t
β = Moving average parameter

4.2.2.11. Train the model:

After preprocessing the input data and defining the model architecture, optimizer, loss-functions activations, etc. We trained our model with training data which is 80% of the total data. The training was done for different choices of hyper-parameters for optimization on the test set.

4.2.2.12. Evaluation Metrics

The performance of the model on a test set, which comprises data that the model hasn't seen during training, is evaluated with the use of evaluation metrics. We may assess how effectively the model can identify false news by comparing the model's predictions with the test set's real labels.

A variety of evaluation metrics are used in our study to determine how well the algorithm can detect fake news. They are:

a) Confusion Matrix

By contrasting the anticipated labels of a piece of data with the actual labels, a classification model's performance is summarized in a table known as a confusion matrix.

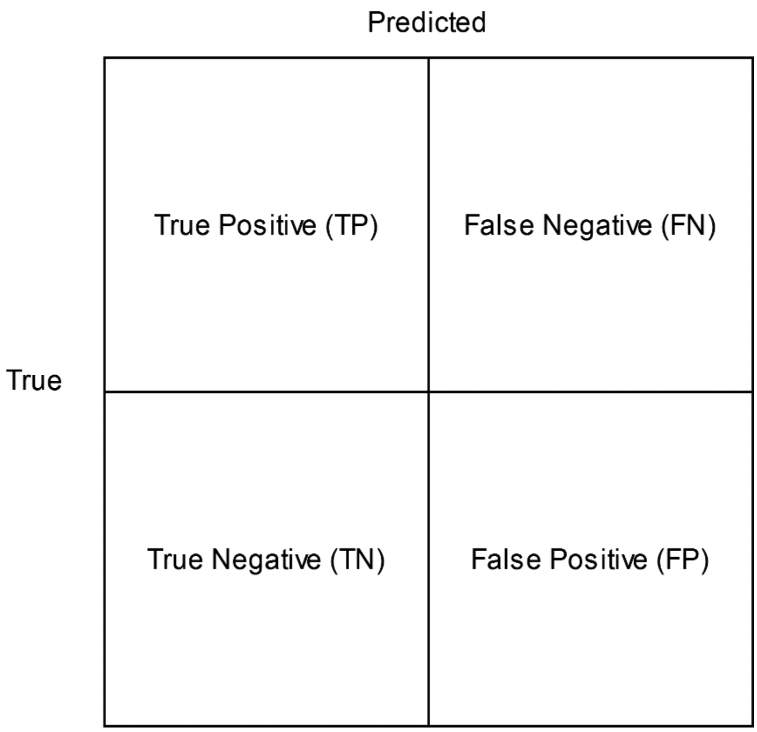


Figure 4-9. Confusion Matrix

The number of cases that the model successfully classified as positive is indicated in the following figure by the abbreviation TP, suggesting that the cases were both positive (in our case fake news) and that the model correctly identified them as such. TN is the proportion of cases that the model correctly recognized as negative, showing that they were indeed negative. FN stands for the number of instances where the model incorrectly categorized positive situations as negative. Lastly, FP denotes the number of instances where the model misclassified a case as positive when it was negative.

b) Accuracy

Among all of the predictions made by the model, accuracy is a performance metric that shows what percentage of them were accurate. It is determined by adding the true positive and true negative results, then dividing by the total number of cases:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (10)$$

c) Precision

The number of cases that were predicted as positive really turned out to be positive depends on precision, which assesses the accuracy of the model's positive predictions. A high precision score means that the model successfully identifies positive cases without misclassifying too many negative examples as positive, and it also means that there are relatively few erroneous positive predictions made by the model.

Mathematically,

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. (11)$$

d) Recall

Recall measures how well a model can distinguish between positive cases from all of the dataset's positive cases. A high recall score shows that the model is properly detecting positive cases while not over-missing positive cases, and it is producing relatively few false negative predictions.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots.. (12)$$

e) F1-score

Precision and recall are balanced by the F1 score by taking into account both false positives and false negatives. It is a helpful parameter for assessing models when we want to strike a balance between precision and recall and keep things from going too far in either direction.

$$F1 - score = \frac{2 \times Precision \ \times recall}{Precision + Recall} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (13)$$

5. IMPLEMENTATION DETAILS

5.1. Web scrapping Method:

This method is used for source verification of social media news. We search the news title on Google and find the similarity of user input news with the results from searching. If we find similar news, then we check if the news is from reliable sources. If both conditions are true then we classify the news as real news.

If the news is not from reliable sources, we cannot say the news is not real. So, we used user Text input and run on a trained BERT model which classifies the news based on its context.

The implementation details of the web scrapping method are listed below:

5.1.1. Google Search:

News title as input taken from the user and google searches it using the python library i.e., **scrape-search-engine**. After searching, news titles and its URL are scraped from the search result. We converted the obtained URL to the base URL using the **urlparse** library the python.

5.1.2. Finding Similarity:

After scraping, the similarity between the input news title and search result news titles is calculated. Cosine similarity is calculated using the BERT model. Bert preprocess and Bert encoder is imported from the tensor-flow library and the news titles are used to calculate embedding vectors. Then cosine similarity is computed with these embedding vectors of input news titles and search result news titles. For calculating Jaccard similarly we use text.split( ) to get individual words. Then Intersection and union are calculated to compute Jaccard similarity. If the cosine similarity and Jaccard similarity were greater than 0.7 and 0.5 respectively then the news is similar.

5.1.3. Source Checking:

Annapurna post, eKantipur, onlinekhabar, gorkhapatraonline, setopati, ratopati, nagriknews, Naya Patrika, and BBC Nepali are a few trustworthy news websites. If there is a resemblance between a specific news headline and the search results, the news source should be investigated. The news can be regarded as dependable and factual if it comes from one of the reputable news websites indicated above.

5.2. BERT Modeling:

5.2.1. Data Collection

For the implementation of this project, we collected 9514 real news from various websites like Gorkhapatra, onlinekhabar, eKantipur, etc. For the fake news dataset, we translated 9514 English fake news data to Nepali using google translate. English fake news dataset was downloaded from Kaggle.

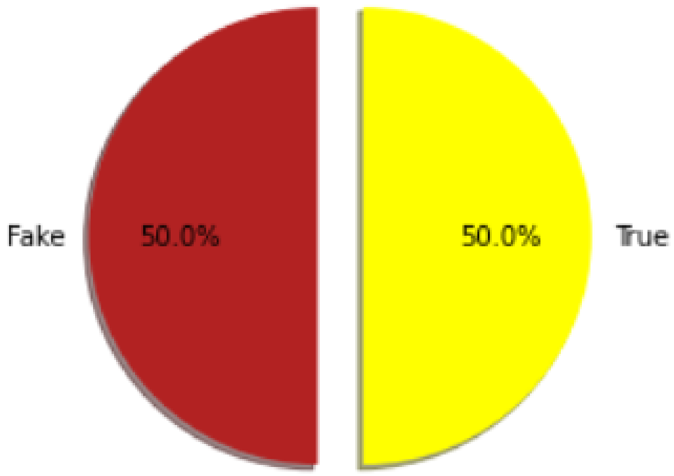


Figure 5-1. Fake and real Data percentage

5.2.2. Data preprocessing

The dataset required for our project is Nepali news in text form. A single news article consists of a title and description. Other data like data sources, authors, and news data are not so relevant for our model since our model is trained to understand the overall context of the text. so, we remove unnecessary columns from our data. Also, for data preprocessing, the following steps are done:

5.2.2.1. Remove Null Values:

True news and fake news datasets are concatenated and labeled as 0 and 1 respectively. For null values removal we use a python library called Pandas. Firstly, check if there is any null value in the dataset, if so, remove complete rows containing the null value using the Pandas dropna() function.

5.2.2.2. Removing Punctuation symbols:

There are different Nepali punctuation symbols. They are unnecessary tokens for model training. So, punctuation symbols have to be removed. To remove them string function str.replace() is used.

'फिल्म ‘भगवत् गीता’मा नायक धिरज मगर अनुबन्धित हुनुभएको छ ।'

फिल्म भगवत् गीतामा नायक धिरज मगर अनुबन्धित हुनुभएको छ

Figure 5-2. Punctuation removal

e.g. str.replace('।', ''), it replaces the ‘|’ punctuation symbol with a blank space.

5.2.2.3. Removing Stop words:

In Nepali text, there are many stop words like ‘उनले’, ‘का’, ‘छ’, etc. which are not important for the meaning of the sentences. So, to remove stop words, the stop words are replaced by blank spaces.

फिल्म भगवत् गीतामा नायक धिरज मगर अनुबन्धित हुनुभएको छ

फिल्म भगवत् गीतामा नायक धिरज मगर अनुबन्धित हुनुभएको

Figure 5-3. Stop words removal

5.2.3. Tokenization and embedding

The model is trained on the description and title of a news article. We used bert_multi_cased_preprocess/3 to generate input mask (for attention), input type–ids, and input word-ids for each text paragraph.

- Input type-ids: It shows the relationship between 2 sentences in a text. It is more useful for next-sentence prediction.

- Input word-ids: They are the integer representation of words. The words are mapped from word to integer and the maximum size of input word ids are fixed regardless of the size of the text: the remaining text is truncated. The maximum size can be set if we use a tokenizer from the transformers library while 128 is for tensorflow. The first token is always CLS (101) and the last token is always SEP(102)

- Input masks: It tells which token to focus on. If the word-id or token is greater than 0, it is set to 1, else set to 0. Its size is the same as the number of tokens in input word ids.

Once the text is tokenized, embedding is done, bert_multi_cased_L-12_H-768-A-12/4 is used from the tensor-flow library is used for this task. This generates vectors that encode the context of tokens. As a result, Encoder output, sequence output, and pooled output are generated.

- Encoder output: Each encoder layer generates a 768-sized vector for each token. There are 12 encoder layers and a fixed size of tokens for each text input. So, one encoder generates multiple vectors for each input text.

- Sequence output: The output resulting from the final encoder layer output is the sequence output. It carries representations of outputs of all the previous layers.

- Pooled output: It is a vector of size 768 for the entire input text. It encodes the context of all token vectors into a single vector. Pooled output is what have used since the context of the entire text as a whole is required for us instead of token-level context. Also, it is computationally efficient for us.

5.2.4. Model Training

The Nepali text is Bert layers' input, which encodes text into vectors. Then the vectors are given as input to the first layer consisting of 768 neurons. The network consists of 3-4 hidden layers and finally a single neuron layer for output prediction.

Dropout regularization has been performed by dropping some percent of neurons randomly in hidden layers. L2 regularization is also added for the weights of the classification layer along with it. This has been done to prevent over-fitting.

5.2.5. Hyperparameter tuning

Different values of hyper-parameters like learning rate, number of epochs, number of hidden layers, percentage of dropout neurons, optimizer, loss function, and activation function were tried out for optimization purposes.

5.2.5.1. Learning Rate

The learning rate hyper-parameter controls how big steps to take while updating the weights of the model with a gradient of the loss function. The model will take a long time to converge to the ideal solution if the learning rate is too small, and it may oscillate or diverge if the learning rate is too high. To train machine learning models for the best performance determining an acceptable learning rate is a crucial task. The learning rate during training can be optimized using a variety of strategies, including learning rate schedules and adaptive learning rate methods. Here, we discovered that the most optimal learning rate was 1e-3.

5.2.5.2. Epoch

When training, an epoch is a single iteration of the complete dataset. The model is trained on the full dataset in batches during each epoch, and its weights are adjusted following the error or loss function. The number of epochs is a hyperparameter that controls how frequently the model iterates through the training dataset. A lower value of the number of epochs can lead to underfitting, where the model is unable to capture the underlying patterns in the data and a higher value of the number of epochs lead to over-fitting, where the model is overly complicated and struggles to generalize to new data. The ideal number of epochs for our model was 30.

5.2.5.3. Batch size

The term "batch size" describes the number of training examples used in a single training iteration. The model is fed a set of training examples with each iteration, and its weights are adjusted based on the error or loss function. The batch size is a hyperparameter that controls how many samples are processed by the model before the weights are updated. The model updates its weights more frequently with a smaller batch size, but this can slow down training and increase the noise in the updates. The model changes its weights less frequently when the batch size is bigger, which can lead to shorter training times and smoother updates but can also result in the model becoming stuck in a local minimum. The batch size we discovered to be the most effective was 32.

5.2.5.4. Activation Function

The output of a node or neuron in the network is determined by the activation function. The network is given non-linearity via the activation function, which transforms the input signal in a non-linear way and introduces non-linearity, enabling the network to learn complex functions or relationships between the input and output. The selection of the activation function is a hyperparameter that affects how well the neural network functions and behaves. Out of different combinations for choices of activations, we used the ReLU for the network's hidden layers, and the output layers use the sigmoid function.

In summary, the optimized hyperparameters for fine-tuning this BERT model are:

- Learning rate = 1e-3
- Epoch = 30
- Batch size = 32
- Activation function = sigmoid and ReLU

After training and optimization, we saved the model for deploying on the web.

5.2.6. Model Deployment

A web application for Nepali Fake News detection was created using Flask, Python, HTML, and CSS and can be run locally on a host machine. HTML and CSS are used to create the User Interface (UI) of the web app and Flask is used as a backend language, to combine UI with the model.

6. RESULTS AND ANALYSIS

6.1. Loss vs Epoch Graph

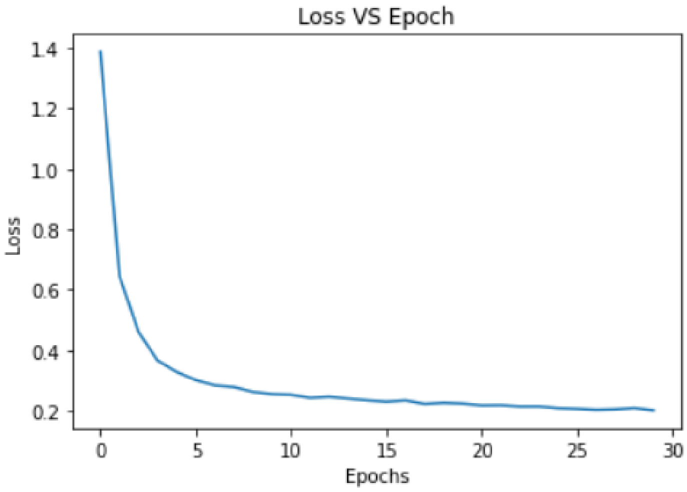


Figure 6-1. Loss vs Epoch Graph

6.2. Accuracy vs Epoch Graph

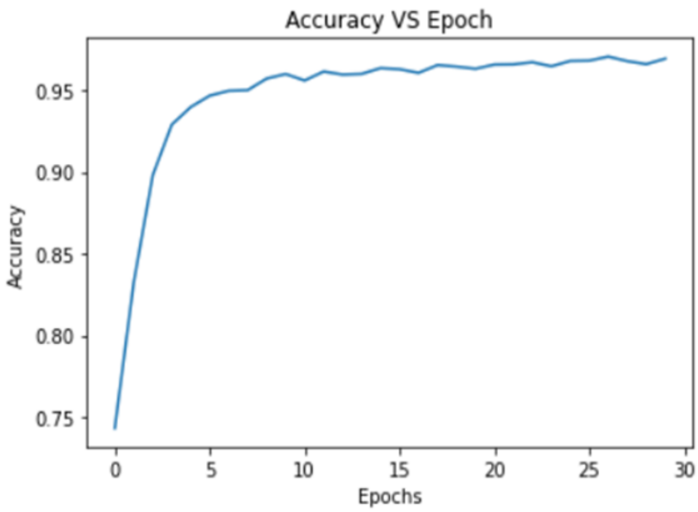


Figure 6-2. Accuracy vs Epoch Graph

6.3. Scatter Plot

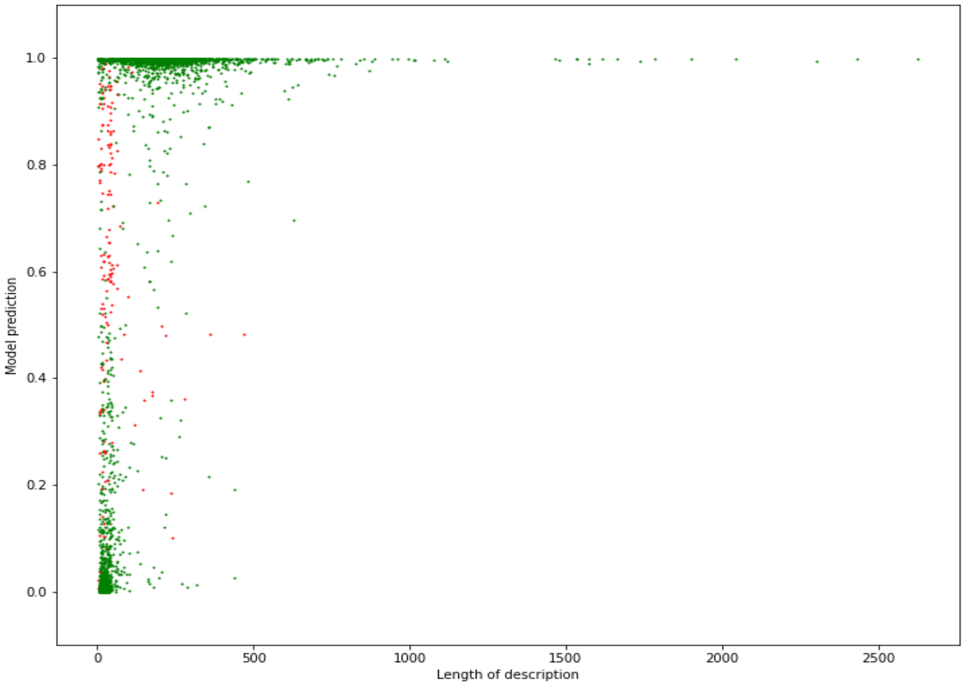


Figure 6-3. Scatter Plot

The x-axis in this graph denotes the length of the news description, and the y-axis is the model forecast. Red color denotes incorrect prediction, whereas green color denotes a good prediction.

6.4. Confusion Matrix

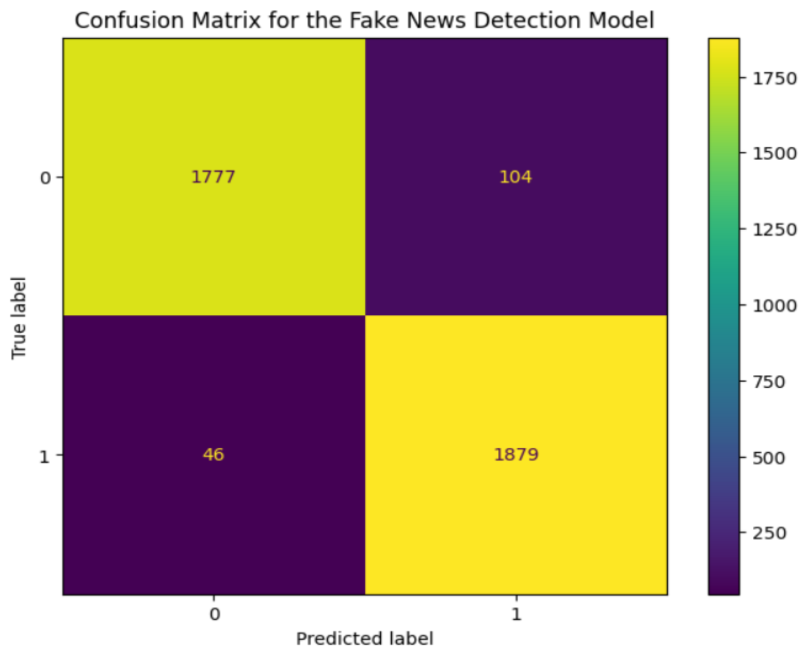


Figure 6-4. Confusion Matrix

From the confusion matrix we can see the following results:

- True Positives (TP) = 1777
- True Negatives (TN) = 1879
- False Positives (FP) = 46
- False Negatives (FN) = 104

6.5. Precision/ Recall vs Threshold Curve

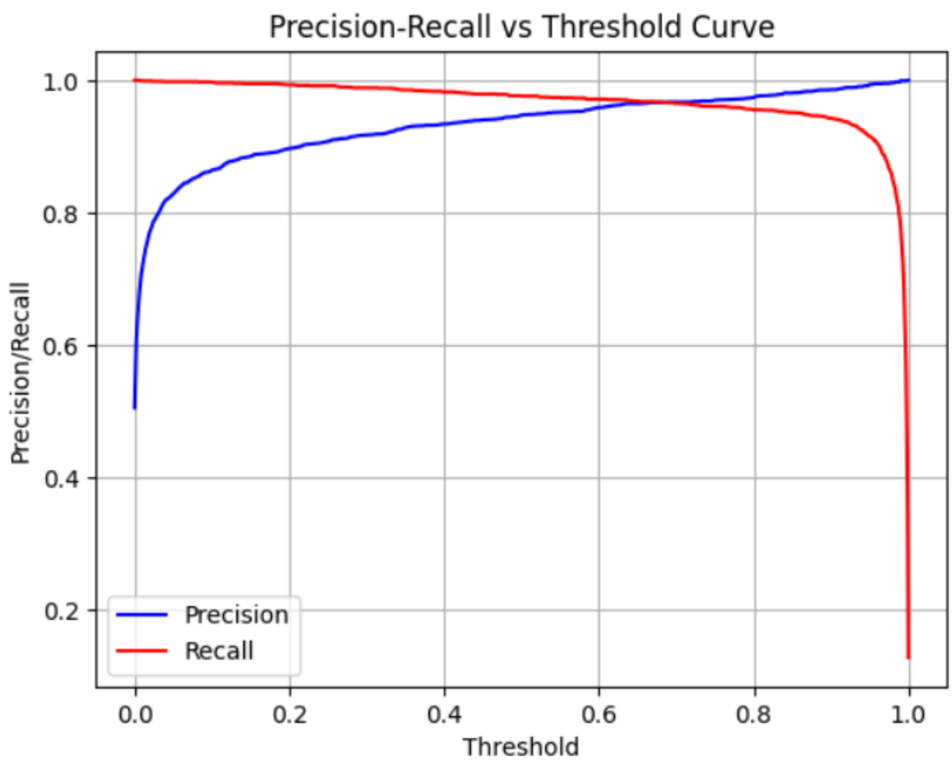


Figure 6-5. Precision/ Recall vs Threshold Curve

6.6. Receiver Opening Characteristic

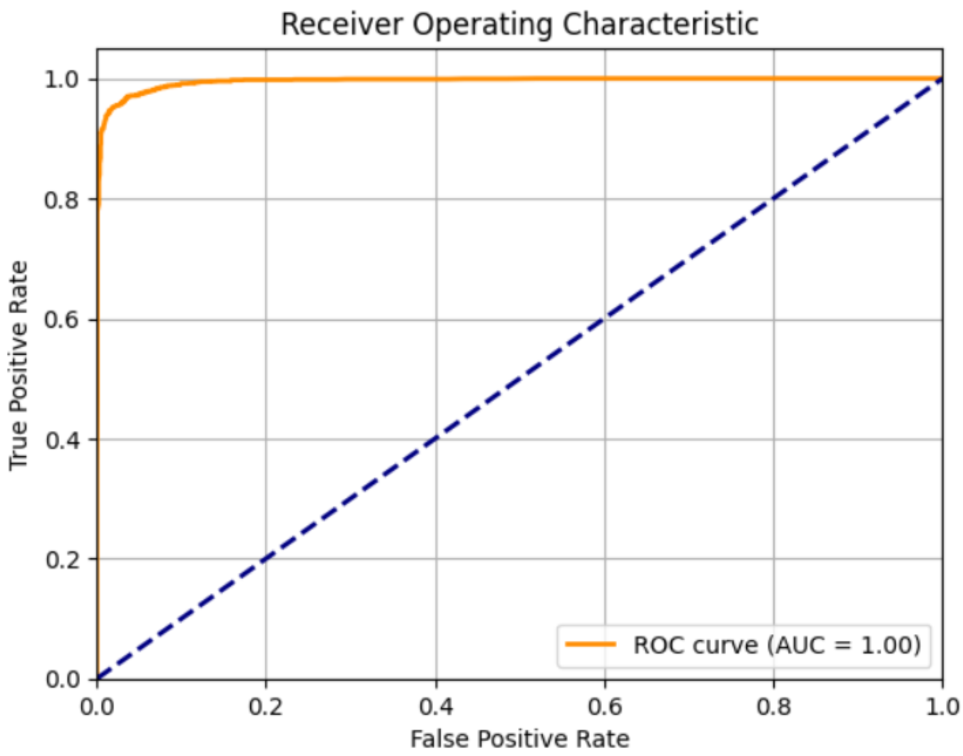


Figure 6-6. Receiver Opening Characteristic

6.7. Evaluation Metrics

Table 6-1. Evaluation Metrics

	precision	recall	f1-score
0	0.97	0.94	0.96
1	0.95	0.98	0.96

0 – True news dataset

1 – Fake news dataset

6.8. UI of website

Figure 6-7. UI of website (part 1)

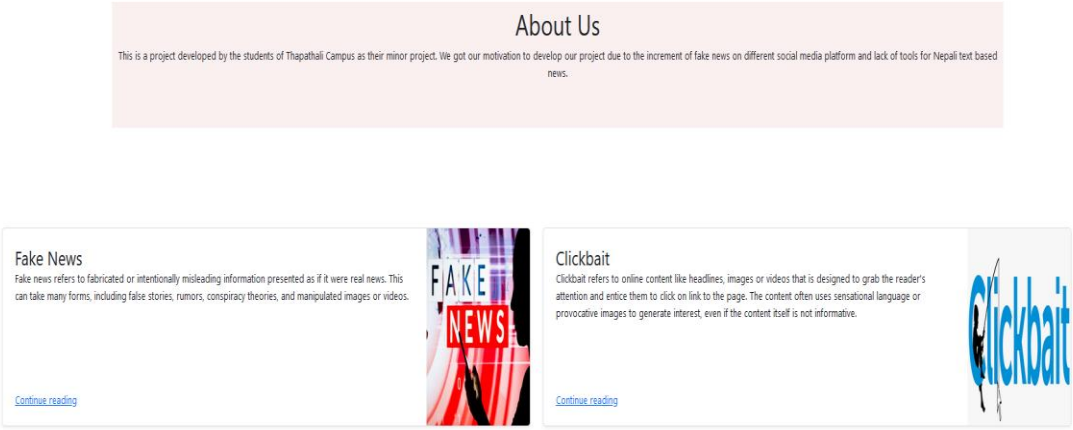


Figure 6-8. UI of website (part 2)

6.9. Home page of the website

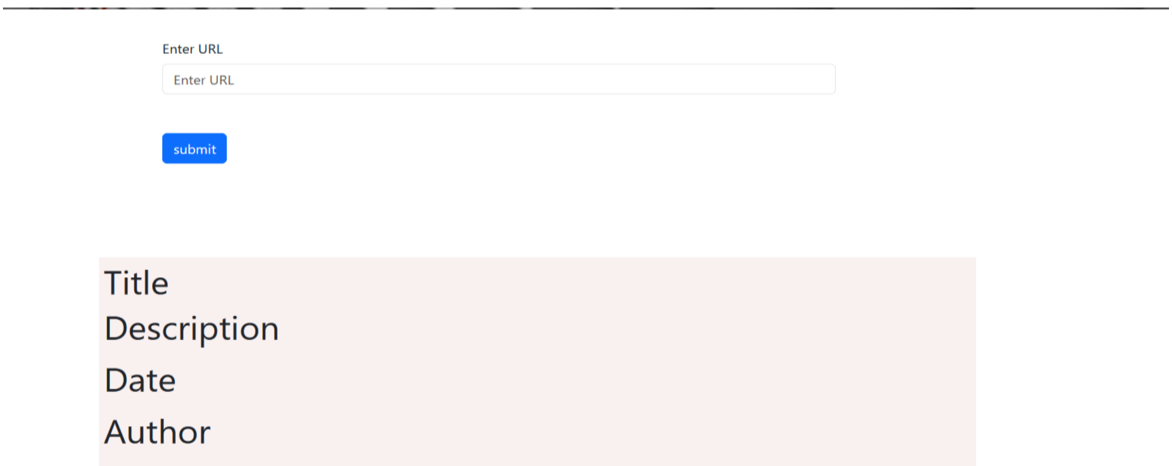


Figure 6-9. Homepage of website

Here, the user can provide the URL of the news website. The title, description, date, and author of the news website are scraped and displayed to the user.

Title

सुत्केरी स्याहार्ने पुर्ख्यौली पेसा

Description

नतिजा हेर्नको लागि लगइन गर्नुहोस् ।तपाईको प्रतिक्रिया समीक्षामा भएकोले प्रकाशित भएको छैन । लुम्बिनी — सुत्केरी र नवजात शिशु स्याहारका लागि तेल मालिस गराउने चलन पुरानै हो । पश्चिम तराईका कतिपय ठाउँमा भने रोजगारीका रूपमा तेल लगाउने र मालिस गर्न हिँड्ने पेसा चलिरहेको छ । यो पेसा विभेदकारी सुनिन्छ, तर ठूला सहरमा 'मसाज' र 'स्पा' का रूपमा सम्मानजनक व्यवसाय बनेकाले गाउँमा यो पेसा गर्नेले पनि आफूलाई स्वाभिमानी ठान्छन् ।धेरैजसो गाउँमा सुत्केरी स्याहार र मालिसका लागि आफ्नै दिदीबहिनी बोलाउने चलन पनि छ । कतै गाउँमा आफन्त, छिमेकीले समेत स्याहार्न सघाइदिने गर्छन् । यही पेसामा लागेका महिलालाई बोलाउने चलन पनि छ । रूपन्देहीको मायादेवी क्षेत्रमा पनि केही महिला पुर्ख्यौली पेसाकै रूपमा सुत्केरी र शिशुलाई मालिस गर्ने काम गर्दै आएका छन् । मायादेवी–८ मतखोलाकी ८३ वर्षीया बन्सराजी हरिजन बुढ्यौली उमेरमा पनि उत्तिकै व्यस्त छिन् । उनलाई दैनिक कसै न कसैले घरमा बोलाइरहेकै हुन्छन् । '५५ वर्षभन्दा बढी भइसक्यो होला मैले काम गरेको,' भारत माइत भएकी उनले भनिन्, 'विवाह भएर आएपछि सासू आमाले सिकाइन्, म त सुँडेनीसमेत भएँ ।' स्वास्थ्यकर्मीको संख्या कम हुने र गाउँमा महिलालाई अस्पताल लैजाने चलन नै नहुँदा आफूले बच्चा निकाल्ने, नाल काट्नेदेखि सुत्केरी स्याहारसुसार गर्ने गरेको उनले सुनाइन् । बिहे गरेर आएपछि सिकेको कामले बाहिरी मान्छेसँग चिनजान, आवतजावत र पारिवारिक सम्बन्ध समेत बढाउन सघाएको बन्सराजीले बताइन् । 'हाम्रो गाउँमा मलाई नचिन्ने कमै मान्छे होलान्,' उनले भनिन्, 'बूढी भए पनि पाकी छ, उसैलाई बोलाउनुपर्छ भनेर लिनै आउँछन् ।'दशकौंदेखि गरिरहेको कामले परिचयसँगै आम्दानीसमेत हुने भएकाले कामप्रति समर्पित रहेको उनको भनाइ छ । 'सुरुसुरुमा हात दुखेर आँसु झर्थ्यो, थकाइ लाग्थ्यो,' उनले भनिन्, 'घरमा सघाइदिने मानिस नहुन्जेलसम्म दुःख भयो, अहिले बानी पर्‍यो ।'१० वर्षअघि बुहारीको निधन भएपछि नातिनातिनाको जिम्मेवारी पनि उनैलाई आयो । त्यसले पनि उनलाई यो उमेरमा काम गरिराख्नु अवस्थामा पुर्‍याएको हो । सुरुमा काम गर्दा ज्यालाबापत थोरै धानचामल दिने गर्थे । अहिले दैनिक ५ सय नघटी सबैले दिन्छन् । 'दुःख नगरे खान लगाउन नपुग्ने त्यसैले सकुन्जेल नछोड्ने भनेर गरिरहेकी छु,' उनले भनिन् । बुढेसकाल लागिसकेपछि भने पहिलेजस्तो जाँगर नचल्ने उनले बताइन् । 'उमेर छँदा हात दरो थियो, अहिले त्यही तागत त आउँदैन,' उनले भनिन्, 'तरिका मिलाएर मालिस गरिदिने भएकाले प्राय मैं बूढीलाई खोज्छन् ।' पहिले पहिले गानो गएमा, घाँटीको गटई बढेको (घाँटीमा फुला आउने) लाई मालिस गरेरै सन्चो बनाइदिने काममा पनि आफूलाई बोलाउने गरेको उनले बताइन् ।५३ वर्षीया भानमती नाउनी छिमेकी पालिकामा समेत पुगेर सुत्केरीलाई तेल लगाउने र मालिस गर्छिन् । आमाबाट मालिसको काम सिकेकी उनले अहिले पनि दैनिक तीन जना महिलासम्मलाई तेल लगाउने गरेको बताइन् । 'महिनाभरि एकै घरमा जाने गरी बोलाएका पनि हुन्छन्,' उनले भनिन्, 'केहीले १० दिन, केहीले १५ दिन आइदेऊ भन्छन् ।' बच्चा जन्माएकी महिलालाई ज्यान दुख्ने भएकाले तेल लगाएर मालिस गरेर आराम दिलाउने चलन नै बसेकाले आफूले पेसाकै रूपमा अँगालिरहेको उनले बताइन् । 'सानै हुँदा आमाले आफू जाने ठाउँमा लगेर पुर्ख्यौली पेसा हो, सिक भन्नुहुन्थ्यो,' उनले भनिन्, 'आमा बित्नुभयो, मैले काम सम्हालिरहेकी छु ।' आफू १२/१३ वर्षकी छँदा आमाले सिकाएर एक्लै काममा पठाउँदा लाज मानेर एकछिनमै घर फर्कने गरेको उनी सम्झन्छिन् । नाउ समुदायको पुर्ख्यौली पेसा नै पुरुषले कपाल काट्ने र महिलाले मालिस गर्दै हिँड्ने रहेको उनले बताइन् । उनका श्रीमान् ज्ञानदासले श्रीमतीलाई सँगै लिएर विवाह, मृत्यु भएका घरमा जाने गर्छन् । विवाह र मृत्यु भएका स्थानमा श्रीमान्ले कपाल काटिदिँदा भानमतीले रङ र हल्दी लगाइदिने गर्छिन् । 'विवाहमा बेहुला–बेहुलीलाई हल्दी र बुकाले मालिस गरेर छाला सफा गराइदिने चलन छ,' उनले भनिन्, 'मृत्यु भएको घरमा १६ दिनमा उम्कने गर्छन् । उम्कंदा रङ लगाएर चोखो बनाइदिने गरिन्छ ।' समुदायअनुसार नाउलाई अघि सारेर दान दक्षिणा वा कपाल काट्ने चलन रहेकाले ती कार्य चल्दा आफ्नो आवश्यकता पर्ने गरेको उनले बताइन् । सुत्केरीलाई मालिस गर्दा एक पटकको अढाई सय कमाउने उनले विवाहमा जाँदा २ देखि ३ हजारसम्म पाउने गरेको सुनाइन् । एक सुत्केरीलाई मालिस गर्दा ४५ मिनेटसम्म लाग्ने गरेको बताइन् ।२ छोरा

Date

2023-03-03 00:00:00

Author

['सन ज', 'प ड ल']

Figure 6-10. Scraped title, description, date, and author from the given URL

6.10. Check news using BERT

Here, user can check if the title is present on the trusted websites or not. If not present on trusted websites, the users can input news descriptions and click on predict button to view the result.

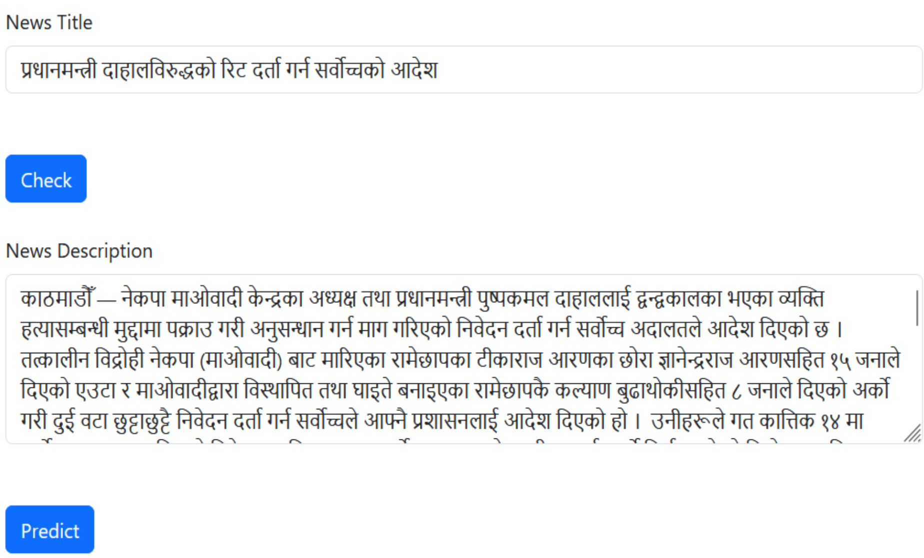


Figure 6-11. Input news title and description

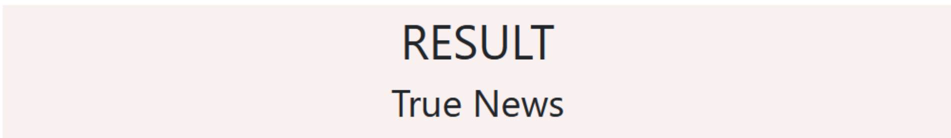


Figure 6-12. Result after prediction

7. FUTURE ENHANCEMENTS

- Address language nuances: Fake news can be intentionally misleading or ambiguous to deceive readers. By addressing language nuances such as sarcasm and irony, fake news detection models could improve their ability to identify fake news.

- Multiple language support: While the current focus of our model is on detecting fake news in a specific language, it may be possible to expand its capabilities to include other languages in the future. This would require collecting and processing a large amount of high-quality data in each respective language, as well as fine-tuning the BERT model for each language.

- Train on Nepali fake news: At present, our fake news detection model has been primarily trained on translated fake news. However, as part of our ongoing efforts to improve the model's accuracy and effectiveness, we plan to collect Nepali fake news and use it to fine-tune the model.

8. CONCLUSION

In conclusion the accuracy and dependability of our system has been good due to the combination of these two methods. The source checking method ensures that user given news articles are true, while BERT model uses semantics to calculate the prediction. BERT has shown to be a useful tool for identifying and categorizing fake news in the Nepali language due to its capacity to capture the context and content of the text. As we did not have actual Nepali fake news in training data, the BERT model's prediction is slightly compromised. By our results we can say that it will surely perform very well if actual fake news was available.

Once model training was finished, the model was able to attain an accuracy of 96%. It was also discovered that fake news had a precision of 95% while real news had a precision of 97%. Also, fake news had a recall score of 94% while real news recall score was 98%. The F1-score for both type of news was 96%. Besides this we checked our system on some real examples and the system showed good results. After that, the model was implemented on a website.

9. APPENDICES

APPENDIX A: Solve GLUE tasks using BERT on TPU

[14]BERT may be used for numerous natural language processing issues. The GLUE benchmark teaches how to optimize BERT for a variety of tasks:

- Is the phrase grammatically correct? — CoLA (Corpus of Linguistic Acceptability).

- Predicting the emotion of a statement is the goal of SST-2 (Stanford Sentiment Treebank).

- See if two phrases are semantically identical using the MRPC (Microsoft Research Paraphrase Corpus).

- Determine whether a pair of questions are semantically identical using QQP (Quora Question Pairs2).

- Given a premise sentence and a hypothesis sentence, the aim is to determine whether the premise implies the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither of the two (MNLI, or Multi-Genre Natural Language Inference) (neutral).

- Question-response Natural Language Inference (QNLI): It is your responsibility to ascertain whether the context sentence responds to the query.

- RTE stands for Recognizing Textual Entailment, which checks if a phrase implies a particular hypothesis or not.

- The goal of WNLI (Winograd Natural Language Inference) is to determine if the original statement and the sentence with the pronoun substitution are implied by one another.

The various operations performed are:

- Loading models from TensorFlow Hub:

  Here, we can select the BERT model from TensorFlow Hub that we want to fine-tune. There are several different BERT models to pick from.

  - The original BERT authors released BERT-Base, Uncased, and seven additional models with training weights.
  - Small BERTs let you experiment with trade-offs between speed, size, and quality since they have the same basic design but feature fewer and/or smaller Transformer pieces.
  - ALBERT: A Lite BERT in four different sizes that share parameters between layers to decrease the model size (but not calculation time).
  - Eight models that are all BERT-based but offer a selection of pre-training domains to better match the objective job are referred to as BERT Experts.
  - Electra gets pre-trained as a discriminator in a setup that mimics a Generative Adversarial Network (GAN) and has the same architecture as BERT (in three different sizes).
  - The Transformer architecture's core features two upgrades thanks to BERT with Talking-Heads Attention and Gated GELU [base, big].

- After this we have to choose a BERT model to fine-tune
- Preprocess the text

  The preprocessing model is employed on the Classify text using BERT Colab and is directly integrated with the BERT encoder.

This article shows how to use a Dataset map to do preprocessing as part of the input pipeline for training, and how to integrate that preprocessing into the model before it is exported for inference. In this approach, despite the TPU's need for numeric inputs, training and inference may be performed with only raw text inputs.

Regardless of TPU needs, asynchronous preprocessing in an input pipeline can improve performance.

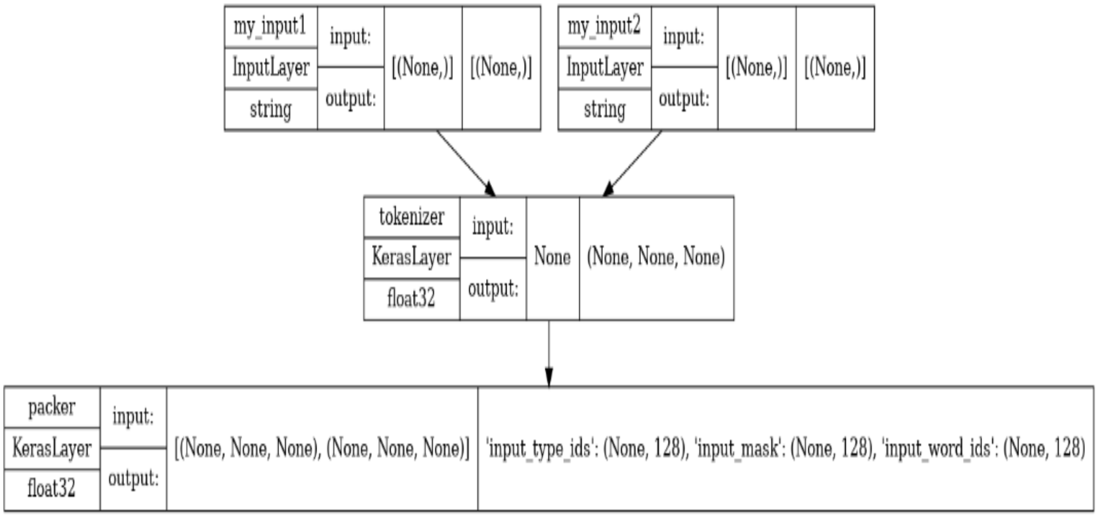


Figure 9-1. Test Process Model of BERT

- Define the Model

  The preprocessed inputs will now be sent through the BERT encoder, a linear classifier will be placed on the top (or another arrangement of layers), and dropout regularization will be used to construct our model for categorizing sentences or phrase pairs.

- Select a task from GLUE.

  We'll apply TensorFlow DataSet from the GLUE benchmark suite.

Because the separate TPU worker host cannot access the local filesystem of the collab runtime, Colab allows us to download these tiny datasets to the local filesystem. The code below reads them totally into memory.

To enable the TPU worker to read data from larger datasets, we'll need to make our own Google Cloud Storage bucket.

The dataset also chooses the suitable loss function for training as well as the issue type (classification or regression).

- Train the model.

Finally, we can fully train the model on the selected dataset.

  - Distribution

    Within the framework of the TPU distribution method, we will develop and construct our primary Keras model and distribute training onto it.

  - Optimizer

    The optimizer setup from BERT pre-training (as in Classify text using BERT) is followed by optimizer fine-tuning: The AdamW optimizer is used, preceded with a linear warm-up phase over the first 10% of training steps (num warmup steps), with a linear decline of a notional starting learning rate. The initial learning rate is lower for fine-tuning in line with the BERT paper (best of 5e-5, 3e-5, 2e-5).

- Export for inference:

The preprocessing component and the newly developed, improved BERT can be included in the final model we produce.

Preprocessing must be integrated into the model at inference time (because there is no longer a separate input queue for training data that does it). Preprocessing is more than simply computation; it also requires the attachment of its resources (the vocabulary table) to the Keras Model that is stored for export. What will be preserved is this vast assemblage.

The model may be saved to Colab, where you can download it later to use it again.

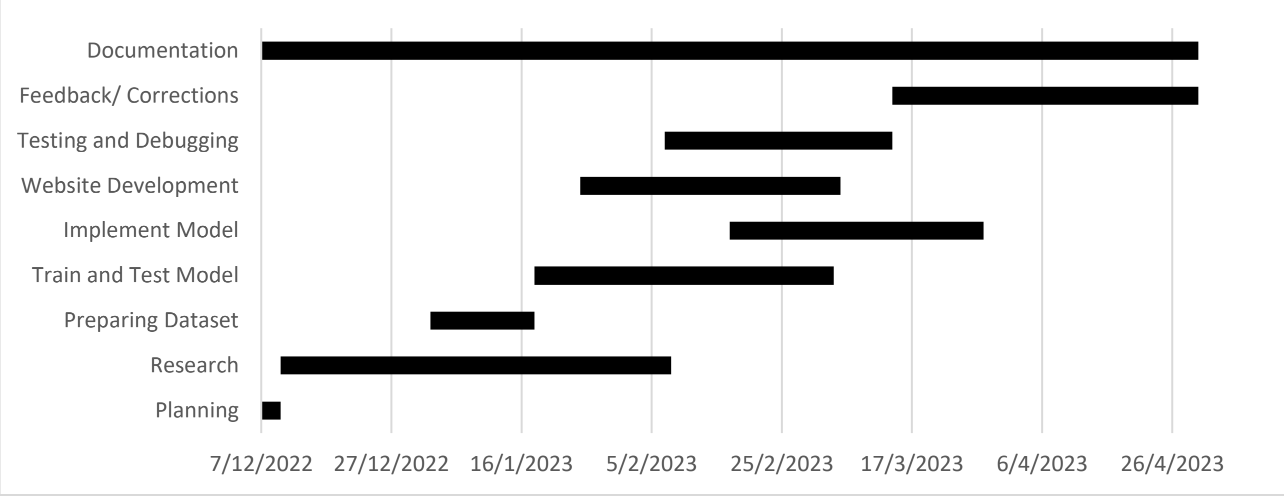
- Testing the Model:

Testing the output of the exported model is the last step.

Reload the model and test it using some inputs from the test split of the dataset to make some comparisons.

APPENDIX B: Gantt-Chart

Table 9-1. Gantt Chart



APPENDIX C: Project Cost

Since all of the libraries used for the implementation of this project are open source there is no such cost required for the implementation of this project.

APPENDIX D: Code Snippets

```
for i in list(true_news):
  true_news[i]=true_news[i].str.replace('|', '')
  true_news[i]=true_news[i].str.replace('?', '')
  true_news[i]=true_news[i].str.replace(':', '')
  true_news[i]=true_news[i].str.replace(';', '')
  true_news[i]=true_news[i].str.replace("'", '')
  true_news[i]=true_news[i].str.replace('"', '')
  true_news[i]=true_news[i].str.replace(',', '')
  true_news[i]=true_news[i].str.replace('.', '')
  true_news[i]=true_news[i].str.replace('(', '')
  true_news[i]=true_news[i].str.replace(')', '')
  true_news[i]=true_news[i].str.replace('\n', '')
  true_news[i]=true_news[i].str.replace('&', '')
  true_news[i]=true_news[i].str.replace('—', '')
  true_news[i]=true_news[i].str.replace('‘', '')
  true_news[i]=true_news[i].str.replace('’', '')
  true_news[i]=true_news[i].str.replace('-', '')
  true_news[i]=true_news[i].str.replace('“', '')
  true_news[i]=true_news[i].str.replace('”', '')
  true_news[i]=true_news[i].str.replace('!', '')
```

Figure 9-2. Punctuation removal

```
from keras.optimizers import Adam
optimizer = Adam(learning_rate=1e-3)

METRICS = [
      tf.keras.metrics.BinaryAccuracy(name='accuracy'),
      tf.keras.metrics.Precision(name='precision'),
      tf.keras.metrics.Recall(name='recall')
]

model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=METRICS)
```

Figure 9-3. Optimizer and loss function

APPENDIX E: Similarity Index

Fake News Detection Using BERT Model

ORIGINALITY REPORT

14%

SIMILARITY INDEX

PRIMARY SOURCES

1	www.researchgate.net Internet	313 words — 3%
2	tensorflow.google.cn Internet	225 words — 2%
3	www.coursehero.com Internet	143 words — 1%
4	deepai.org Internet	59 words — < 1%
5	web.archive.org Internet	58 words — < 1%
6	flipkarma.com Internet	50 words — < 1%
7	I. Kadek Sastrawan, I. Putu Agung Bayupati, Dewa Made Sri Arsa. "Detection of fake news using deep learning CNN-RNN based methods", ICT Express, 2021 Crossref	49 words — < 1%
8	doctorpenguin.com Internet	46 words — < 1%
9	www.csustan.edu Internet	45 words — < 1%

10	www.mdpi.com Internet	45 words — < 1%
11	"Understanding COVID-19: The Role of Computational Intelligence", Springer Science and Business Media LLC, 2022 Crossref	39 words — < 1%
12	nanopdf.com Internet	33 words — < 1%
13	Dhiren Rohera, Harshal Shethna, Keyur Patel, Urvish Thakker, Sudeep Tanwar, Rajesh Gupta, Wei-Chiang Hong, Ravi Sharma. "A Taxonomy of Fake News Classification Techniques: Survey and Implementation Aspects", IEEE Access, 2022 Crossref	32 words — < 1%
14	assets.researchsquare.com Internet	32 words — < 1%
15	mdpi-res.com Internet	32 words — < 1%
16	elibrary.tucl.edu.np Internet	25 words — < 1%
17	nzkled.agropodlipami.pl Internet	22 words — < 1%
18	medium.com Internet	21 words — < 1%
19	"Encyclopedia of Social Network Analysis and Mining", Springer Science and Business Media LLC, 2014 Crossref	20 words — < 1%

20 downloads.optimumg.com
Internet
20 words — < 1%

21 elibrary.tucl.edu.np:8080
Internet
18 words — < 1%

22 towardsdatascience.com
Internet
18 words — < 1%

23 kth.diva-portal.org
Internet
17 words — < 1%

24 "Advances in Computing and Data Sciences", Springer Science and Business Media LLC, 2019
Crossref
16 words — < 1%

25 Abdulla All Noman, Mustafizur Rahaman, Tahmid Hasan Pranto, Rashedur M. Rahman. "Blockchain for medical collaboration: A federated learning-based approach for multi-class respiratory disease classification", Healthcare Analytics, 2023
Crossref
16 words — < 1%

26 instituto.perfil.com
Internet
16 words — < 1%

27 www.ijres.org
Internet
16 words — < 1%

28 "Advances in Information Retrieval", Springer Science and Business Media LLC, 2020
Crossref
15 words — < 1%

29 Sungmin Yun, Carlos H. Caldas. "Analysing decision variables that influence preliminary feasibility studies using data mining techniques", Construction Management and Economics, 2009
14 words — < 1%

Crossref

30	research.spit.ac.in Internet	13 words — < 1%
31	Jiawei Zhang, Bowen Dong, Philip S. Yu. "FakeDetector: Effective Fake News Detection with Deep Diffusive Neural Network", 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020 Crossref	12 words — < 1%
32	Yu Zhang, Jinyue Shi, Dongliang Wang, Chengxin Pang, Zhuangzhuang Yang, Baizheng Wu, Yabing Xu, Weihua Du. "Detection on Safety Helmet Wearing of Distribution Network Construction Based on YOLOv5-Btri Algorithm", 2022 2nd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 2022 Crossref	12 words — < 1%
33	1library.net Internet	10 words — < 1%
34	Mu-Yen Chen, Yi-Wei Lai. "Using Fuzzy Clustering with Deep Learning Models for Detection of COVID-19 Disinformation", ACM Transactions on Asian and Low-Resource Language Information Processing, 2022 Crossref	10 words — < 1%
35	diglib.tugraz.at Internet	10 words — < 1%
36	journalofbigdata.springeropen.com Internet	10 words — < 1%
37	ukdiss.com Internet	10 words — < 1%

38 www.rroij.com
Internet

10 words — < 1%

39 Ahmed J Abougarair, Mohammed Elmaryul, Mohamed KI Aburakhis. "Real time traffic sign detection and recognition for autonomous vehicle", International Robotics & Automation Journal, 2022
Crossref

9 words — < 1%

40 Devisha Arunadevi Tiwari. "RMCL: A deep learning based recursive malicious context learner in social networks", Computational Intelligence, 2022
Crossref

9 words — < 1%

41 Palwinder Kaur, Amandeep Kaur. "Classification of Breast Cancer Mammographic Images Using A Light-Weighted Convolutional Neural Network", 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), 2023
Crossref

9 words — < 1%

42 Priti Shaw, Suresh Sankaranarayanan, Pascal Lorenz. "Early Esophageal Malignancy Detection Using Deep Transfer Learning and Explainable AI", 2022 6th International Conference on Communication and Information Systems (ICCIS), 2022
Crossref

9 words — < 1%

43 Soroush Vosoughi, Mostafa 'Neo' Mohsenvand, Deb Roy. "Rumor Gauge", ACM Transactions on Knowledge Discovery from Data, 2017
Crossref

9 words — < 1%

44 files.osf.io
Internet

9 words — < 1%

45	umpir.ump.edu.my Internet	9 words — < 1%
46	vocal.media Internet	9 words — < 1%
47	www-poc.dell.com Internet	9 words — < 1%
48	www.linkreducation.com Internet	9 words — < 1%
49	"Machine Learning, Image Processing, Network Security and Data Sciences", Springer Science and Business Media LLC, 2020 Crossref	8 words — < 1%
50	Hamdi Béji, Toufik Kanit, Tanguy Messager. "Prediction of Effective Elastic and Thermal Properties of Heterogeneous Materials Using Convolutional Neural Networks", Applied Mechanics, 2023 Crossref	8 words — < 1%
51	IFoA Publications	8 words — < 1%
52	Pritha Mitra, Lija Jacob. "Fake News Detection and Classify the Category", 2022 International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT), 2022 Crossref	8 words — < 1%
53	Shivam Singh, Manish Kumar Goyal. "An innovative approach to predict atmospheric rivers: Exploring convolutional autoencoder", Atmospheric Research, 2023 Crossref	8 words — < 1%

54 Yang Liu. "Depression clinical detection model based on social media: a federated deep learning approach", Research Square Platform LLC, 2023 — 8 words — < 1%
Crossref Posted Content

55 Zhan Li. "A schema for ontology-based concept definition and identification", International Journal of Computer Applications in Technology, 2010 — 8 words — < 1%
Crossref

56 academic.oup.com — 8 words — < 1%
Internet

57 d197for5662m48.cloudfront.net — 8 words — < 1%
Internet

58 dspace.mist.ac.bd:8080 — 8 words — < 1%
Internet

59 smartech.gatech.edu — 8 words — < 1%
Internet

60 stars.library.ucf.edu — 8 words — < 1%
Internet

61 ugspace.ug.edu.gh — 8 words — < 1%
Internet

62 www.ijraset.com — 8 words — < 1%
Internet

63 www.preprints.org — 8 words — < 1%
Internet

64 www.repo.uni-hannover.de — 8 words — < 1%
Internet

65	www.ukessays.com Internet	8 words — < 1%
66	Amgad Muneer, Shakirah Mohd Taib, Suliman Mohamed Fati, Abdullateef O. Balogun, Izzatdin Abdul Aziz. "A Hybrid Deep Learning-Based Unsupervised Anomaly Detection in High Dimensional Data", Computers, Materials & Continua, 2022 Crossref	6 words — < 1%
67	Oliver Faust, Alex Shenfield, Murtadha Kareem, Tan Ru San, Hamido Fujita, U. Rajendra Acharya. "Automated detection of atrial fibrillation using long short-term memory network with RR interval signals", Computers in Biology and Medicine, 2018 Crossref	6 words — < 1%
68	Sulove Bhattarai, Sudip Bhujel, Santosh Adhikari, Shanta Maharjan. "Design and Implementation of a Portable ECG Device", Journal of Innovations in Engineering Education, 2020 Crossref	6 words — < 1%
69	Viktar Atliha. "Improving image captioning methods using machine learning approaches", Vilnius Gediminas Technical University, 2023 Crossref Posted Content	6 words — < 1%

EXCLUDE QUOTES	ON	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF

References

[1] C. Elliott, "Forbes," 21 February 2019. [Online]. Available: https://www.forbes.com/sites/christopherelliott/2019/02/21/these-are-the-real-fake-news-sites/?sh=2554c7c23c3e.

[2] D. Adhikari, "Onlinekhabar," 13 December 2022. [Online]. Available: https://english.onlinekhabar.com/misinformation-elections-nepal.html.

[3] L. Breiman and A. Cutler, *Random Forests*, 2001.

[4] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Google AI Language, 2019.

[5] K. Stahl, "Fake news detection in social media," 2018.

[6] J. Zhang, B. Dong and P. S. Yu, "FAKEDETECTOR: Effective Fake News Detection," 2019.

[7] G. K, S. Ramani and M. K. S, "Identification of Fake News through SVM and Random Forest," 2020.

[8] T. Yesugade, S. Kokate, S. Patil, R. Varma and S. Pawar, "Fake News Detection using LSTM," International Research Journal of Engineering and Technology (IRJET), 2021.

[9] U. Sharma, S. Saran and S. M. Patil, "Fake News Detection using Machine Learning Algorithms," International Journal of Engineering Research & Technology (IJERT), 2021.

[10] I. K. Sastrawan, I. Bayupati and D. M. Sri, "Detection of fake news using deep learning CNN–RNN based methods," 2021.

[11] M. Farokhian, V. Rafe and H. Veisi, "Fake news detection using parallel BERT deep neural networks," 2022.

[12] S. R. Dam, S. P. Panday and T. B. Thapa, "Detecting Clickbaits on Nepali News using SVM and RF," 2021.

[13] N. Shakya and P. Poudyal, "Detection of fake news using deep neural networks," *Kathmandu University Journal of Science, Engineering and Technology,,* vol. 16, no. 2, 2022.

[14] "Tensorflow," [Online]. Available: https://www.tensorflow.org/text/tutorials/bert_glue.