TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARMENT OF ELECTRONICS AND COMPUTER ENGINEERING

BALKHU, KATHMANDU

A Major Project Report On

"Sentiment Analysis Using Deep Neural Network for Nepali Language"

CT755

Submitted by-

Om Bikram Hamal [25632]

Prashanna Khanal [25641]

Prativa Shah [25643]

A project report submitted to the department of Electronics and Computer Engineering in the Partial fulfillment of the requirements for degree of Bachelor of Engineering in computer Engineering

Kathmandu, Nepal

Supervised by:

Er. Manish Mallick

April, 2022

ADVANCED COLLEGE OF ENGINEERING AND MANAGEMENT

DEPARTMENT OF COMPUTER AND ELECTRONICS ENGINEERING

LETTER OF APPROVAL

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "**Sentiment Analysis Using Deep Neural Network for Nepali Language"** submitted by

Om Bikram Hamal [25632]

Prashanna Khanal [25641]

Prativa Shah [25643]

In partial fulfillment for the Bachelor's degree in Computer Engineering.

Supervisor

Er. Manish Mallick

External Examiner

Bibha Sthapit

Er. Ajaya Shrestha

Head of Department Senior Lecturer

Department of Electronics and Computer Engineering

Date:

COPYRIGHT

ACKNOWLEDGEMENT

We take this opportunity to express my deepest and sincere gratitude to our Supervisor **Er. Manish Mallick** , Lecturer, Department of Electronics and Computer Engineering for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project and also for his constant encouragement and advice throughout our Bachelors program.

We express our deep gratitude to **Er. Ajay Shrestha**, HOD**, Er. Bikash Acharya**, DHOD, **Er. Bigyan Karki**, Project Co-Ordinator, **Er. Lochan Lal Amatya,** principal**,** Department of Electronics and Computer Engineering for their regular support, co-operation, and co-ordination.

The in-time facilities provided by the department throughout the Bachelor's program are also equally acknowledgeable.

We would like to convey our thanks to the teaching and non-teaching staff of the Department of Electronics & Communication and Computer Engineering, Acem for their invaluable help and support throughout the period of Bachelor's Degree. We are also grateful to all our classmates for their help, encouragement and invaluable suggestions.

Finally, yet more importantly, I would like to express my deep appreciation to my grandparents, parents, sister and brother for their perpetual support and encouragement throughout the Bachelor's degree period.

Om Bikram Hamal [25632]

Prashanna Khanal [25641]

Prativa Shah [25643]

ABSTRACT

With the abundance of Nepali Unicode text in internet due to social media like; Twitter, Facebook and Nepali news portal/websites, people express their feelings, opinions through text by writing reviews about products, movies, news, ongoing situation etc. Analysis of opinion for product, news or document could be beneficial to many companies, institutions and individuals for marketing, advertising, question answering, product selection and so on. This project aims to retrieve and pre-process the data form social media, news portal, movie review, product feedback for sentiment analysis, which is a part of natural language processing. This system shows various approaches taken to build Nepali language sentiment classification system using Recurrent Neural Network which classifies the sentiment polarity of a given sentence into “positive”, and “negative”. The system takes input as Nepali sentence and gives the sentiment class like positive or negative as an output.

Keywords: *classification, sentiment, abundance, polarity, neural, recurrent.*

TABLE OF CONTENT

Title **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

RNN	Recurrent Neural Network
NLTK	Natural Language Toolkit
DFD	Data Flow Diagram
API	Application Program Interface
NLP	Natural Language Processing
POS	Part of Speech
RT	Retweet
LSTM	Long Short Term Memory
Bi-LSTM	Bi-Directional Long Short Term Memory
RegEX	Regular Expression

CHAPTER 1

INTRODUCTION

1.1 Background

Sentiment analysis is becoming a popular study these days, mainly because social networking sites include online users who are free to express their thoughts, feelings and impressions concerning a specific topic. In fact, nowadays, any kind of marketing business is currently immersing to the new trends of businesses. Apart from written surveys, the companies also extend their customer satisfaction analysis through the web, in order to gather a large amount of data. Few studies on sentiment analysis have already been presented. These studies are targeted to Twitter, for tweet updates about a specific topic, mostly on brands of products.

These systems collect raw data from twitter, using hash tags, like #example topic, and use the data as a corpus to be feed upon implementing the classifying method. However, gathering and analyzing data from the social networking site like Twitter has one known downside. Every twitter update is restricted to 140 characters in length. For this reason, Twitter users tend to use heavy abbreviations and fragmented expressions. The social networking site Facebook will be the targeted website for this project. This is because Facebook, unlike Twitter, has 5000 characters for every status update. For this reason, a clearer sentence construction would be more expectable. Moreover, the number of Facebook users is abundant, namely it is a good sample for creating a corpus. Concerning that sample, the determination of the polarity of the people's opinion would be quite interesting. Sentiment analysis is involved in the study of opinion mining.

Furthermore, sentiment analysis is commonly used by advertisers, movie creators and other organizations that wish to acquire their customers' reaction on a specific topic. Although the simplest way to gather opinions is in the form of surveys, there are few drawbacks, which consist of great handicaps of the marketing research. The problems emerging of this approach are the conduct of a survey for each product or feature, the format, the distribution and timing of the survey, and the reliance on the good will of people to take the survey. All the above-mentioned problems need high maintenance for the marketing research group's view. Opinion mining in sentiment analysis also faces few challenges for the system developers' perspective. In the case

of opinions, not all words used in the sentence have significance. Some words are classified as noise because they are of no use in the process of classifying the polarity of the opinion. Also, there are words like “not” and whenever they are added to a positive word, they will attach a negative meaning to the existent opinion. Aside from words, symbols like sad face “/” or a happy face “☺” present significance in natural language processing. Hence, there are not only the words, which take place in the observation procedure.

1.2 Motivation

Aspect-based Sentiment Analysis assists in understanding the opinion of the associated entities helping for a better quality of a service or a product. A model is developed to detect the sentiment in Nepali text using Recurrent Neural Network algorithm like LSTM, Bi-LSTM. Therefore, instead of spending times in reading and figuring out the positivity or the negativity of text we can use automated techniques for sentimental analysis.

1.3 Problem Statement

As motivated by the rapid growth of text data, text mining has been applied to discover hidden knowledge from text in many applications and domains. In business sectors, great efforts have been made to find out customers’ sentiments and opinions, often expressed in free text, towards companies’ products and services. However, discovering sentiments and opinions through manual analysis of a large volume of textual data is extremely difficult. Hence, in recent years, there have been much interests in the natural language processing community to develop novel text mining techniques with the capability of accurately extracting customers’ opinions from large volumes of unstructured text data.

Among various opinion mining tasks, one of them is sentiment classification, Sentiment analysis is a type of data mining that measures the inclination of people’s opinions through natural language processing (NLP), computational linguistics and text analysis, which are used to extract and analyze subjective information from the web - mostly social media and similar sources. The analyzed data quantifies the general public's sentiments or reactions toward certain products, people or ideas and reveal the contextual polarity of the information. Sentiment analysis is also known

as opinion mining. Sentiment analysis carries the basic task of classification of the expressed opinion in a document into “positive” and “negative”.

In recent years, many Nepali online news portals like onlinekhabar.com, ratopati.com, setopati.com have been providing news online in Nepali Unicode. Similarly, people use Nepali Unicode to comment, review such online news. Not much work about Nepali Natural Language Processing has been found during our research. Since the Nepali language is morphologically rich and complex the text classifier needs to consider specific language features before classifying the text. Preprocessing of data in Nepali Unicode is at a delicate stage. Very low resources about Nepali Unicode are available.

The applications for sentiment analysis are endless. More and more we’re seeing it being used in social media monitoring and VOC to track customer reviews, survey responses, competitors, etc. However, it is also practical for use in business analytics and situations in which text needs to be analyzed. Sentiment analysis is mostly used to create recommendation systems which are user specific. The opinion of a user about a specific product is the key factor for determining the likes and dislikes of the user which results in increase of efficient of traditional recommender systems. Today’s consumers buy products recommended by Amazon, watch movies recommended by Netflix, and listen to music recommended by Pandora. Research showed that the recommendation system developed using sentiment analysis of user reviews which was used by Amazon, increased their profit boost by 29 percent. Similar results can be achieved in the Nepali market by using a sentiment analyzer for Nepali language.

1.4 Project Objective

- To present a model that can analyze the Nepali text using deep neural network architecture.

1.5 Significance of the study

We perform Sentiment Analysis on Nepali text. We believe that it is possible to more accurately classify the emotion/sentiment in Nepali text. Nepali language (text) is very complex and diversity so that it will be very difficult analyze sentiment in Nepali text. are more succinct than reviews and are easier to classify. Their ability to contain more character allows for better writing and a more accurate portrayal of emotion. This will help to promote company brand on social media. Also, it will help to design marketing strategy in Nepal. Build a presence, extent engagement and measuring your return on investment. Beside this some other significances are:

- It is used to find the opinion of the customers on the newly lunched products.
- It is used in the film industry to analysis the comments and reviews of the movies and songs.
- It is used in News Portals to find out the public opinions and reviews.
- It is used in Politics to know the political perceptions and views of the people.

CHAPTER 2
LITERATURE REVIEW

There are lots of research which have been done in the field of Sentiment Analysis, but there are few research works on Sentiment Analysis based on Nepali texts.

Birat Bade Shrestha, Bal Krishna Bal (2020)[1] have implemented Named-Entity Based Sentiment Analysis of Nepali News Media Text. Their architecture consisted of Data Collection, Preprocessing, Named Entity Recognition, Anaphora Resolution and Sentiment Analysis. For the data collection step, news articles were scraped from four online news portals. Collected datasets were cleaned via article cleaning subcomponents and further lemmatized to split word into root forms and to remove unwanted suffixes under Preprocessing step. POS tagging, NER tagging and Lappin and Leass algorithm for anaphora resolution were performed on the collected datasets to increase the overall accuracy of the system. Word2Vec and FastText were used for feature extraction using Skip Gram parameters. Due to low datasets the sentiment analysis was trained on Support Vector Machine, Decision Tree and Random Forest where Support Vector Machine had better overall score. SVM had F1-score of 80.2 and 78.7 on Word2Vec and FastText respectively. Total of 3490 sentences were scraped for this research method out of which 2676 datasets were positive and 814 were negative.

Abbi Nizar Muhammad, Saiful Bukhori, Priza Pandunata (2019)[2] worked on Sentiment Analysis of Positive and Negative of YouTube Comments Using Naive Bayes – Support Vector Machine (NBSVM) Classifier. The combination of Naïve Bayes and Support Vector Machine produced better accuracy level and stronger performance with the use of a 7:3 scale of data that is 70% training data and 30% testing data. By producing the highest performance test values, namely precision of 91%, recall of 83% and f1 score of 87%.

Tej Bahadur Shahi, Ashok Kumar Pant (2018)[3] performed Nepali news classification using Naive Bayes, Support Vector Machines and Neural Networks. This research evaluates some most widely used machine learning techniques, mainly Naive Bayes, SVM and Neural Networks, for automatic Nepali news classification problem. To experiment the system, a self-created Nepali News Corpus with 20 different categories and total 4964 documents, collected by crawling different online

national news portals, is used. TF-IDF based features are extracted from the preprocessed documents to train and test the models. The average empirical results show that the SVM with RBF kernel is outperforming the other three algorithms with the classification accuracy of 74.65%. Then follows the linear SVM with accuracy of 74.62%, Multilayer Perceptron Neural Networks with accuracy 72.99% and the Naive Bayes with 68.31% accuracy.

Lal Bahadur Reshmi Thapa, Bal Krishna Bal (2016)[4] worked on Classifying sentiments in Nepali subjective texts [4]. In this work, they looked into applying three Machine Learning classifiers, namely Support Vector Machine, Multinomial Naive Bayes and Logistic Regression for developing a model to classify book and movie reviews written in Nepali into “Positive” and “Negative”. They evaluated and validated their model using 5-fold cross-validation techniques. Experimental results showed that the Multinomial Naive Bayes classifier performs with a higher accuracy than the other two classifiers.

CHAPTER 3

REQUIREMENT ANALYSIS

3.1 Software Requirements

Software required for this application are:

- Windows OS 7 or higher
- Python
- Visual studio code
- Flask

3.2 Hardware Requirements

Hardware requirement to run this system is PC with internet connection.

3.3 Functional Requirements

- Sentiment analysis of Nepali text.

3.4 Non-Functional Requirements

These requirements are not needed by the system but are essential for the better performance of sentiment engine. The points below focus on the non-functional requirement of the system.

- **Reliability**

  The system is reliable. Sentiment prediction matches 80% of the time.

- **Maintainability**

  A maintainable system is created, and Sentiment Analyzer Engine is able to train on new input data and is scalable to millions of data points.

- **Performance**

  The forward pass from the neural network is a fast process. For the engine, fast matrix computation occurs.

- **Portability**

  Sentiment Analyzer engine is portable, and it is easy to integrate into any web application or mobile application imaginable by the use of the REST APIs made.

3.5 Feasibility Study

The following points describes the feasibility of the project.

- **Technical Feasibility**

  The software is to be developed using deep learning models which can easily be implemented NLP sectors for analysis of sentiment in Nepali text. Also, the team members have sufficient programming and related knowledge which will enable us to learn and adapt to these specific languages and platforms easily. Thus, we can see that the project is technically feasible.

- **Economic Feasibility**

  The program uses programming languages whose IDEs are freeware. The remaining cost is that of training the developer team in the particular language and/or platform, which is minimal. So, the project is economically feasible.

- **Operational Feasibility**

  The software requires very little specific environment to run. The software will be extremely user-friendly, removing the need for specifically trained employees. Similarly, the cost of buying the rights and the maintenance cost will not be very high for the client. So, the software is feasible for operation.

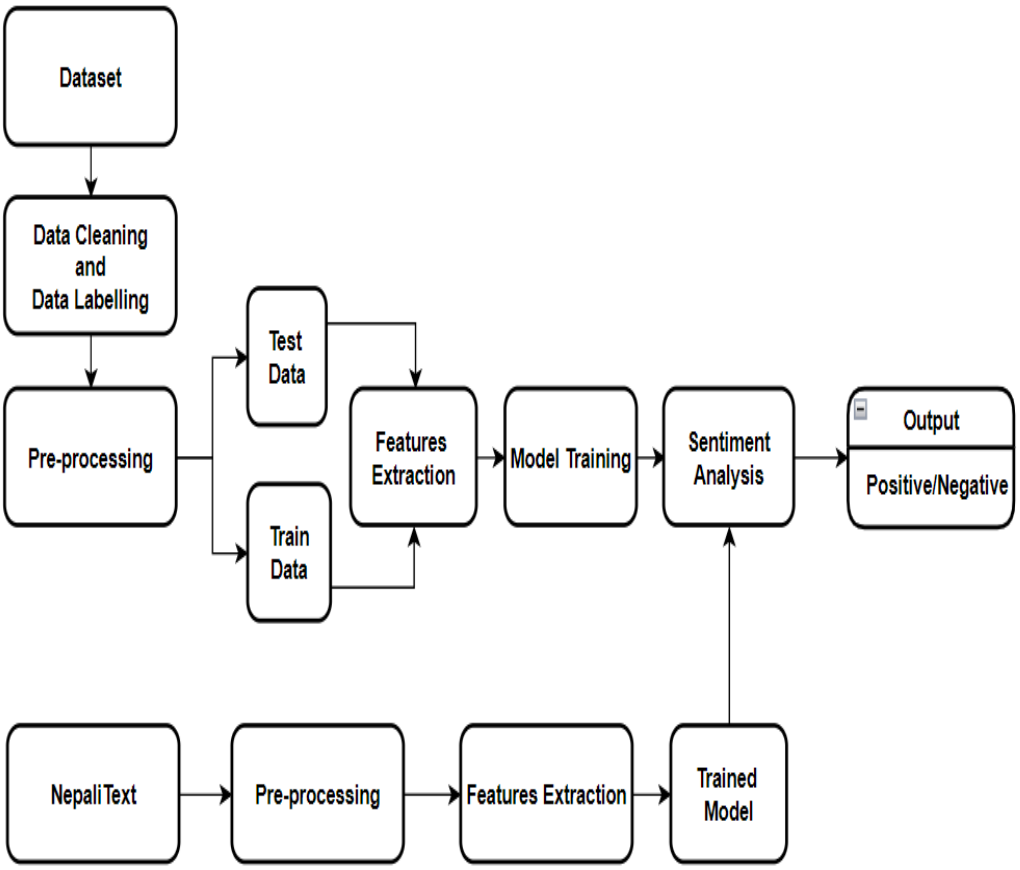
CHAPTER 4

SYSTEM DESIGN AND ARCHITECTURE

4.1 System Design

Training data: From the preprocessed data, we selected 80% of the data for training model.

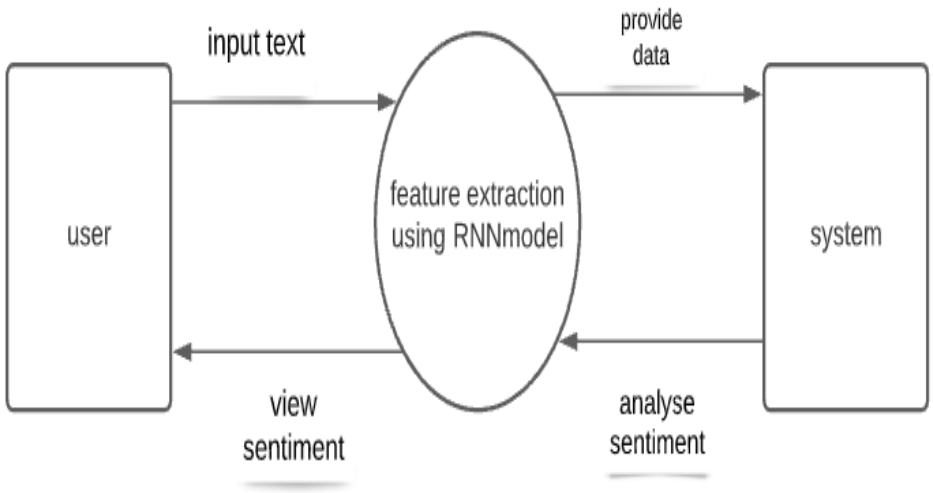
Testing data: The remaining 20% data are used as testing data. Now after getting the dataset, we need to preprocess the data and provide labels to each of the text given there during training the dataset. First, the input datasets are trained. To extract features from RNN model first we need to train the RNN network.



4.1 System Design

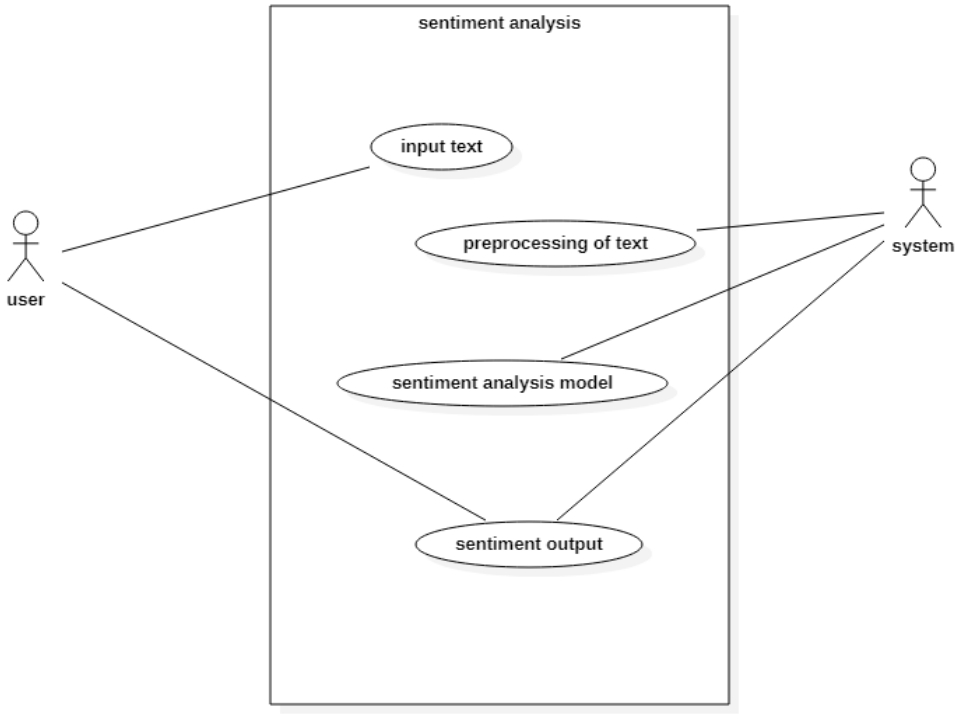
4.2 Data Flow Diagram

DFD level 0



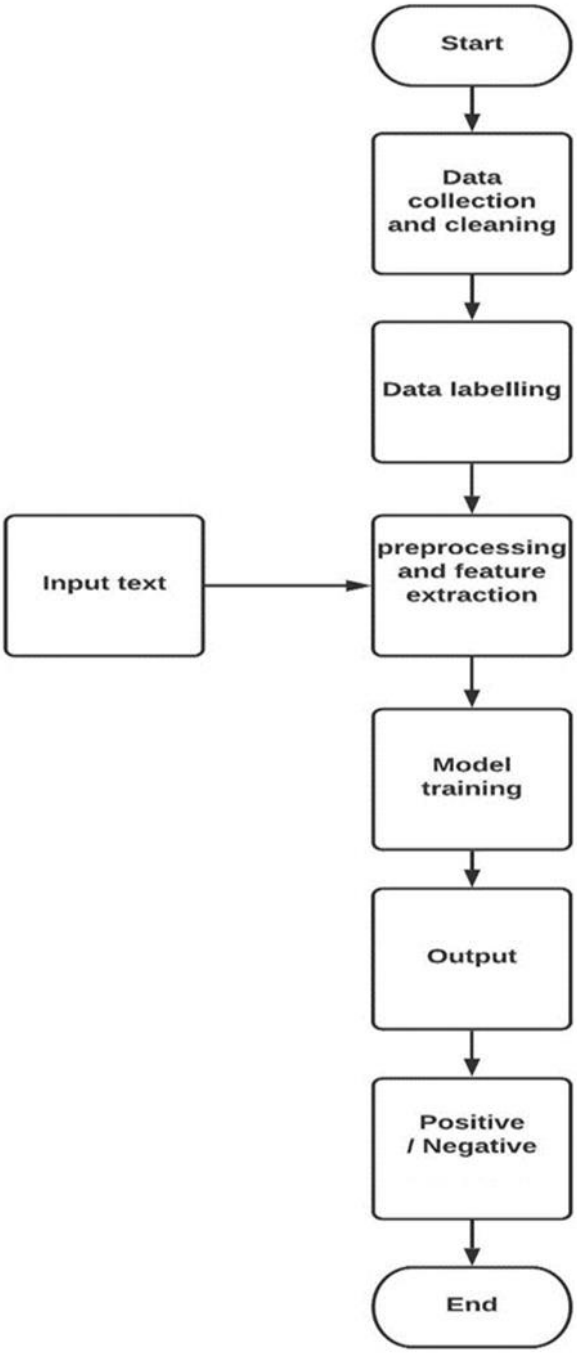
4.2 Level 0 DFD

4.3 Use Case Diagram



4.3 Use Case Diagram

4.4 System Flowchart



4.4 System Flowchart

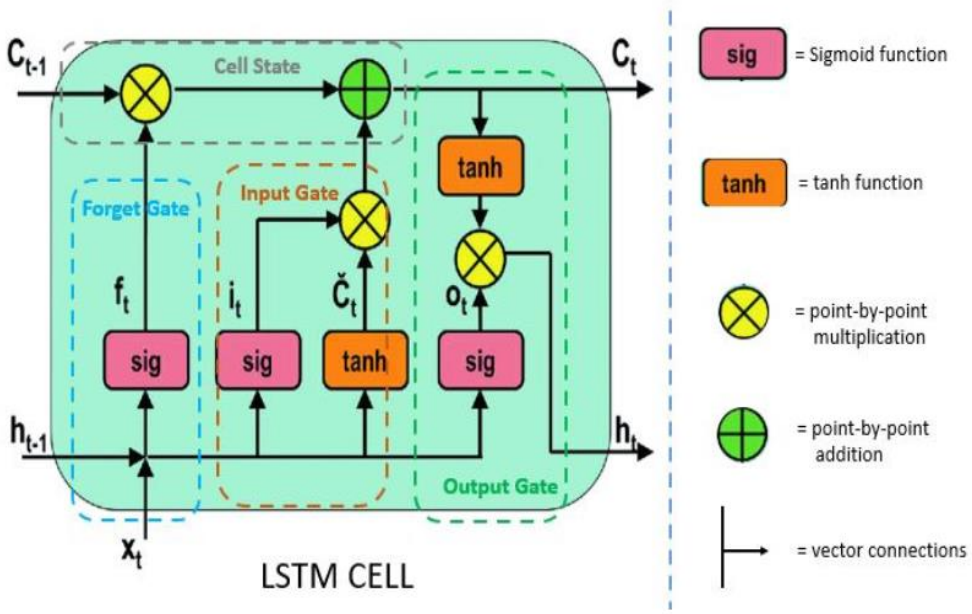
CHAPTER 5

METHODOLOGY

5.1 Algorithms

5.1.1 LSTM RNN

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. In addition, this type of network is better for maintaining long-range connections, recognizing the relationship between values at the beginning and end of a sequence. It trains the model by using back-propagation. In LSTM we have three gates:



5.1.1 Lstm Gates

[Source: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn]

1. Input gate: Discover which value from input should be used to modify the memory. Sigmoid function decides which values to let through 0, 1 and tanh function gives weightage to the values which are passed deciding their level of importance ranging from -1 to 1**.**

2. **Forget gate**: Discover what details to be discarded from the block. It is decided by the sigmoid function. it looks at the previous state(h_t-1) and the

content input (X_t) and outputs a number between 0(omit this) and 1(keep this) for each number in the cell state C_t-1.

3. Output gate: The input and the memory of the block is used to decide the output. Sigmoid function decides which values to let through 0,1 and tanh function gives weightage to the values which are passed deciding their level of importance ranging from -1 to 1 and multiplied with output of Sigmoid.

4. Tanh: Tanh is a non-linear activation function. It regulates the values flowing through the network, maintaining the values between -1 and 1. To avoid information fading, a function is needed whose second derivative can survive for longer. There might be a case where some values become enormous, further causing values to be insignificant.

5. Sigmoid: Sigmoid belongs to the family of non-linear activation functions. It is contained by the gate. Unlike tanh, sigmoid maintains the values between 0 and 1. It helps the network to update or forget the data. If the multiplication results in 0, the information is considered forgotten. Similarly, the information stays if the value is 1.

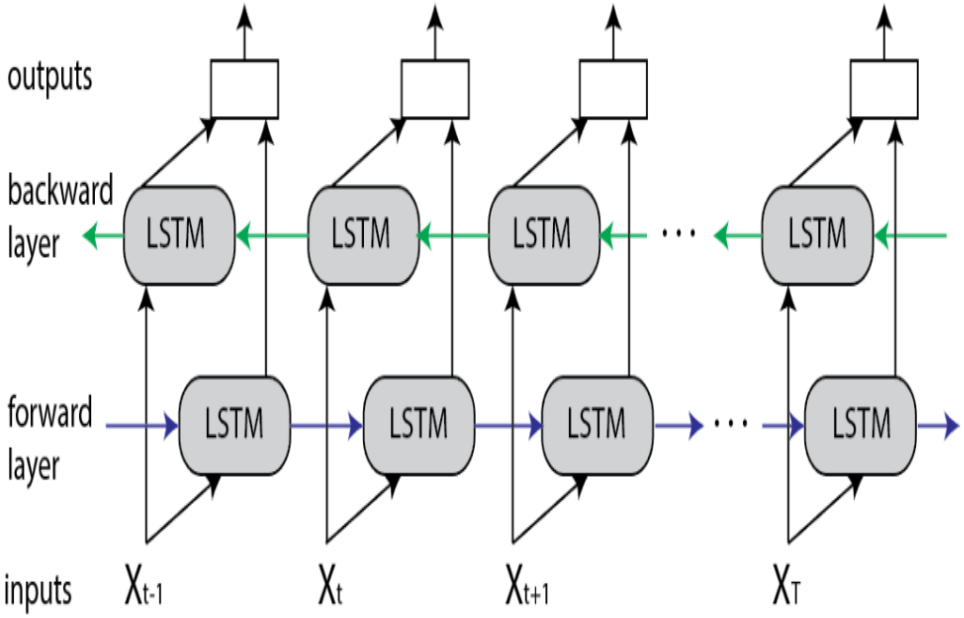
5.1.2 Bi-LSTM RNN

Bidirectional LSTM (Bi-LSTM) is a recurrent neural network used primarily on natural language processing. Unlike standard LSTM, the input flows in both directions, and it's capable of utilizing information from both sides. It's also a powerful tool for modeling the sequential dependencies between words and phrases in both directions of the sequence.

In summary, Bi-LSTM adds one more LSTM layer, which reverses the direction of information flow. Briefly, it means that the input sequence flows backward in the additional LSTM layer. Then we combine the outputs from both LSTM layers in several ways, such as average, sum, multiplication, or concatenation.

This type of architecture has many advantages in real-world problems, especially in NLP. The main reason is that every component of an input sequence has information from both the past and present. For this reason,

BiLSTM can produce a more meaningful output, combining LSTM layers from both directions.



5.1.2 Bi-LSTM Network Architecture

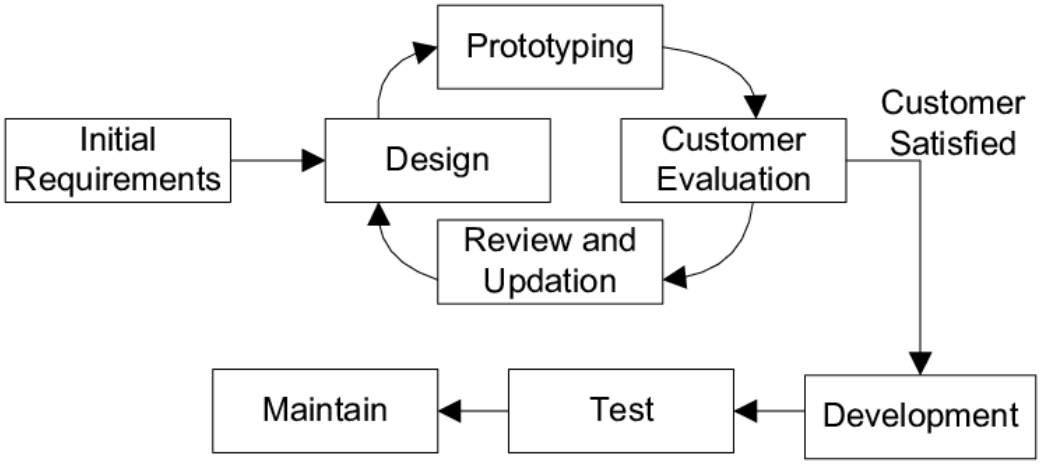
[Source: https://www.baeldung.com/cs/bidirectional-vs-unidirectional-lstm]

BiLSTM will have a different output for every component (word) of the sequence (sentence). As a result, the BiLSTM model is beneficial in some NLP tasks, such as sentence classification, translation, and entity recognition. In addition, it finds its applications in speech recognition, protein structure prediction, handwritten recognition, and similar fields.

5.2 Software Development Model

5.2.1 Prototyping model

Prototyping model is the model of software development life cycle where the Iterative Process starts with a simple implementation of the software requirements and iteratively enhances the evolving versions until the full system is implemented.



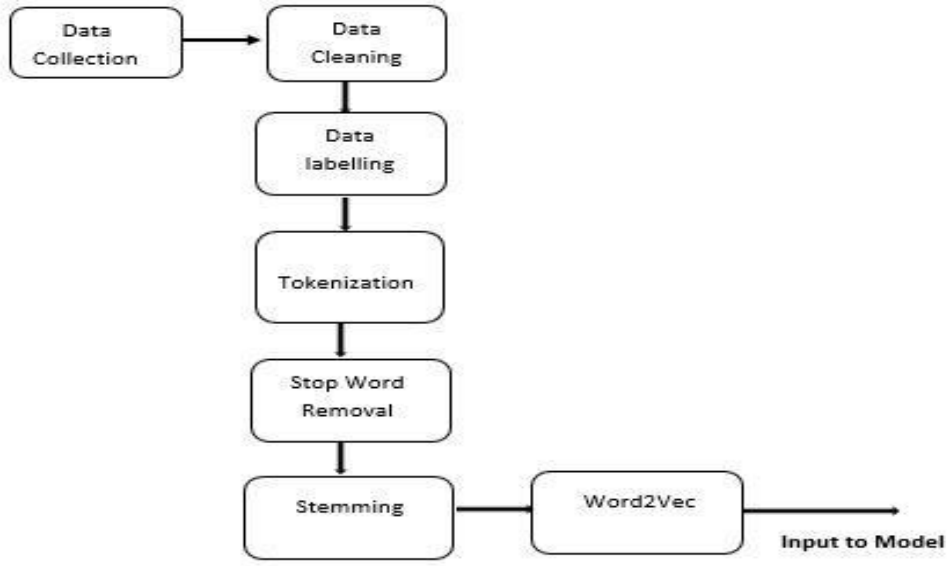
5.2.1 Prototype Model

[Source: https://www.researchgate.net/figure/Prototype-Model]

Here, the developer and client interact to establish the requirements of the software. The essence of prototyping is a quickly designed and can undergo immediate evaluation. Here, the visible elements of the software, the input and the output are designed. The final product of the design through this model is a prototype. After the prototype is developed, the client evaluates the prototype and provides its recommendations and suggestion to the developer. Until the all the user requirements are met, it continues in an iterative manner. Hence this model accommodates problem of changing requirements.

5.3 Pre-Processing

For the better performance of classifiers, some transformations and removal of certain contents were done before processing. With the help of NLTK library preprocessing is performed in input data set. After the data has been collected, it was processed to make into correct format so that the neural networks can understand both the inputs and outputs. The following block diagram shows the data preparation steps of the project.



5.3 Pre-Processing Steps

5.3.1 Dataset Collection

We have collected the data from twitter, using web-scraping technology also from twitter tweet feed of popular Nepali tweet pages. The data is collected programmatically from the website as different websites have different designs and architectures. The data is mainly collected from the opinions section of the site as the section contains more opinions to mine to give more insightful data and making labeling easy. The API for social networking websites were also used.

For example, we applied for Twitter Developer Account to get API keys to gather data from twitter. Tweepy ,which is a python tweet API library was used to make API requests and model the response in appropriate format. We used some of the queries for the API like; ‘brb1954’, ‘hello_sarkar’, ‘thapagk’, ‘nepalitweet’ etc.

From these queries the tweets were returned which then was tokenized into sentences and modeled finally saving into CSV format.

5.3.2 Data cleaning

To make dataset clean and labelled noises were removed by the system such as unnecessary punctuations, emojis, numbers and other symbols. Also removed the English words, mentioned and RT(Re-tweets).and labelled the data manually. For the cleaning process, we use NLTK library called as Regular Expression (RegEx).

5.3.2 Regular Expressions with Description

SN.	RegEx	Description
1	`/@[a-zA-Z0-9_]*/um`	Replace twitter handles with white spaces.
2	`/[।|?]*/um`	Remove full stops and question marks in Nepali Script because of our system working on sentence level classification.
3	`/[a-zA-z!]/um`	Remove all English characters (a-z and A-Z) and exclamations.
4	`/\.\.\./um`	Remove trailing full stops at the end of truncated tweets.
5	`/[:] /um`	Remove ':' characters present in tweets.
6	`/#/um`	Remove hashtags in tweet.

5.3.2 Data Cleaning

5.3.3 Dataset Labelling

We have labelled more than 3500 datasets manually. We perform the individual evaluation after labelling the dataset. We remove the sarcastic types of sentences while labelling.

निखिल दाइ को कुरा एक्दमै पोजेटिभ कुरा गर्नु भयो सलुट छ दाइ	positive
हाहाहाहा हँसायो कुन पुत्रिकार हो यो दोसि ले न्यायपाउनु पर्छ भन्ने डाम्ना	negative
प्रस्न सोद्ने को गोरु हो	negative
रवि जि नेपाली दुखी र गरिब जनता को हरु को आईडल हुन जय रवि	positive
कति खेर हो आज रबि सर रिहा हुने उहा को मुहार हेर्ने आतुर भा को छ	positive

5.3.3 Dataset Labelling

5.3.4 Tokenization

The unlabeled and unprepared data is fed into the tokenization engine. There are two types of tokenization: one sentence level tokenization and another word level tokenization. There is the use of Regular Expressions also known as RegEx for splitting the documents into tokens. The algorithm for sentence level tokenization is given below:

a. Start

b. Input Nepali Sentences

c. RegEx Split Sentences (“|”)

d. Output split chunks of Nepali Sentences

e. End

In languages that have words separated by blank spaces, the token boundary for word level tokenization is the blank/white space. Nepali is one such language so word-level tokenization in Nepali can be achieved by stripping tokens at those white spaces. However, it is not as simple as it looks, especially when dealing with punctuations. Some punctuation is easy to handle. We first need to replace punctuations with white spaces. Similarly, hyphen (-) which is used in linking word pairs: opposite, analogy or similar, together. In this case hyphen is considered as the part of token itself. However, hyphen might occur independently in such cases we need to attach the words to make it a single token. Similarly, colon (:) and period (.) can be considered as the part of token itself. The algorithm for word level tokenization can be seen below:

Word Level Tokenization Algorithm

a. Start

b. Input Sentence

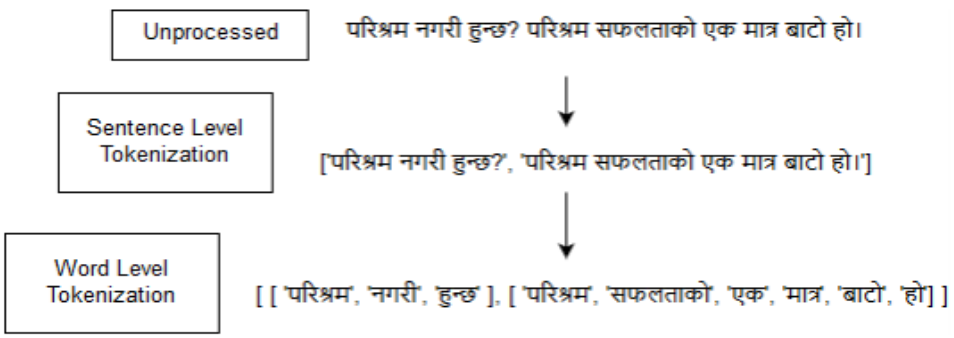
c. Replace punctuations with white spaces

d. Make all words with (:, ., -) symbols a single token

e. Split the formed sentences according to white spaces

f. Output the tokenized sentences

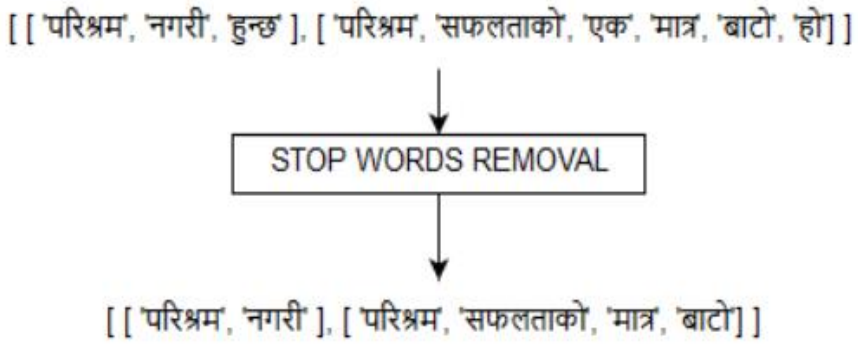
g. End



5.3.4 Tokenization Process

5.3.5 Stop Words Removal

The tokenized Nepali words will now be preprocessed to remove stop words such as छ, हो etc. which has no meaning or participation in sentiment analysis. This stop word removed tokenized data will now be fed into stemming algorithm.



5.3.5 Stop Words Removal Process

5.3.6 Stemming

We will use snowball rule-based stemming algorithm for removing some of the stems of the Nepali language such as को, का, कक etc.

The stemming algorithm was used in our project to reduce the redundancy imposed by various words coming from the same stems meaning similar definition for this already implemented algorithm in snowball was used. Snowball is a small string processing language which was designed for creating stemming algorithms that is used in information retrieval tasks. The language was created for creating readily available stemming algorithms and easy information retrieval.

Snowball stemming algorithm was used in our project which is the implementation of Shrestha, I., & Dhakal, S.S. (2016)[9] which is suffix stripping in nature. The algorithm

was created from 128 suffix rules which are executed step-by-step and in iterative

manner to eliminate inflections in Nepali Language. The stemmer was tested in 5000 Nepali words to give overall accuracy around 88.78% words. The following algorithm shows simplified flowchart about how the snowball stemming algorithm works for three different types of suffixes in the word.

Suffix Category I:

a. Start

b. Input Words containing category I

c. Strip the suffix

d. Scan for Category II and III

e. If suffix found then go to respective stemming processes else store

f. Stop

Suffix Category II:

a. Start

b. Input words containing Category II

c. Check stripping criteria

d. If satisfied then scan for category II and III and go to respective processes else store the word

e. Stop

Suffix Category III:

a. Start

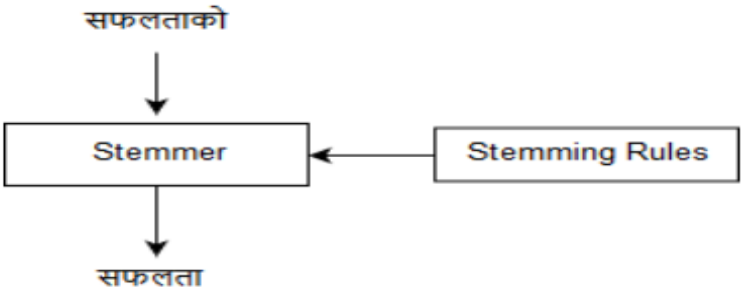
b. Input words containing category III

c. Strip the suffix

d. Scan for Category II and III

e. If found then go to respective stemming processes else store

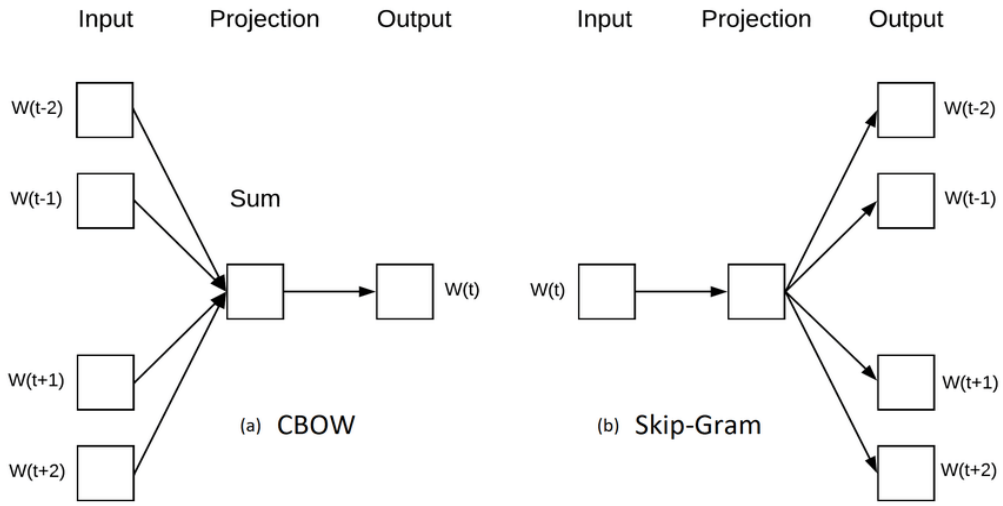
f. Stop



5.3.6 Stemming Process

5.3.7 Word2Vec

As neural networks require numbers as the inputs. The tokenized words need to be converted into feature vectors i.e. list of numbers. Not any random list of number but the related numbers. There must be a correlation between two tokenized words i.e. two vectors must be correlated with each other. Example: In English language, king and boy are more related than queen and girl so king and boy's vector representation should be near to each other than other words. Word2Vec converts those sentences into context linked vector representation as the name suggests.



5.3.7 Word2Vec Model Architecture

[Source: https://arxiv.org/pdf/1301.3781.pdf]

1. Continuous Bag of Words (CBOW) Model

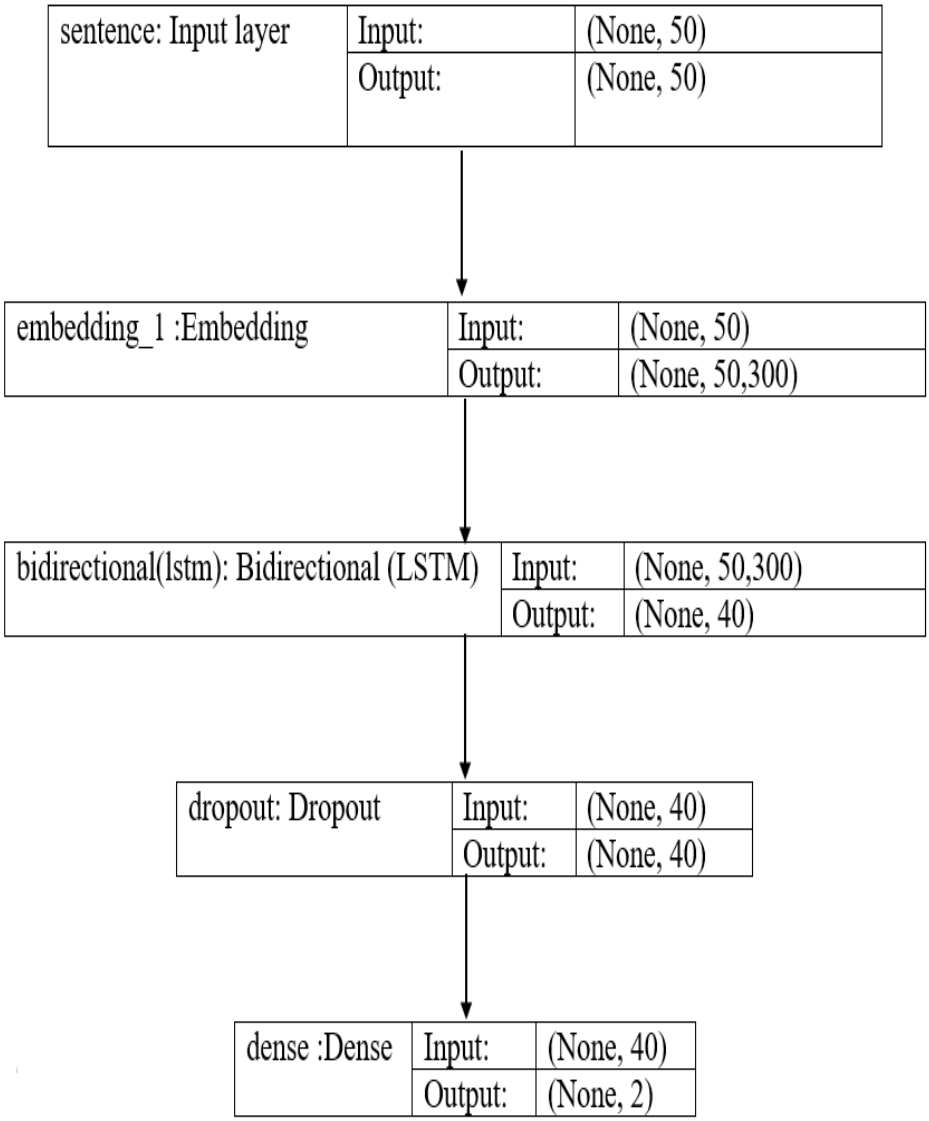
In the continuous bag of words architecture, the model predicts the current word from the surrounding context words. The length of the surrounding context word is the window size that is a tunable hyperparameter. The model can be trained by a single hidden layer neural network.

2. Skip-Gram Model

In the skip-gram model, the neural network is trained to predict the surrounding context words given the current word as input. Here also the window size of the surrounding context words is a tunable parameter. When the neural network is trained, it produces the vector representation of the words in the training corpus. Hera also the size of the vector is a hyperparameter that can be experimented with to produce the best results.

5.3.8 Classification Model

For classification, we use one layers of Bi LSTM to learn the task. We use the bidirectional LSTM structure for training the sentiment classification model which is a wrapper around LSTM layers that enables bidirectional data flow in the sequences during training process. This model was trained for 4 minutes in vs code for 3 epochs with the training accuracy of 83% and validation accuracy of 80%. The model used for training and inference is shown by the figure below alongside the hyperparameters taken to train the model.



5.3.8 Classification Model

CHAPTER 6

RESULTS AND ANALYSIS

6.1 Model Training

We trained our model with 3 epochs, batch size= 32, training sample = 2880 and testing sample= 720. We got the training accuracy around 83% and testing accuracy around 80%.

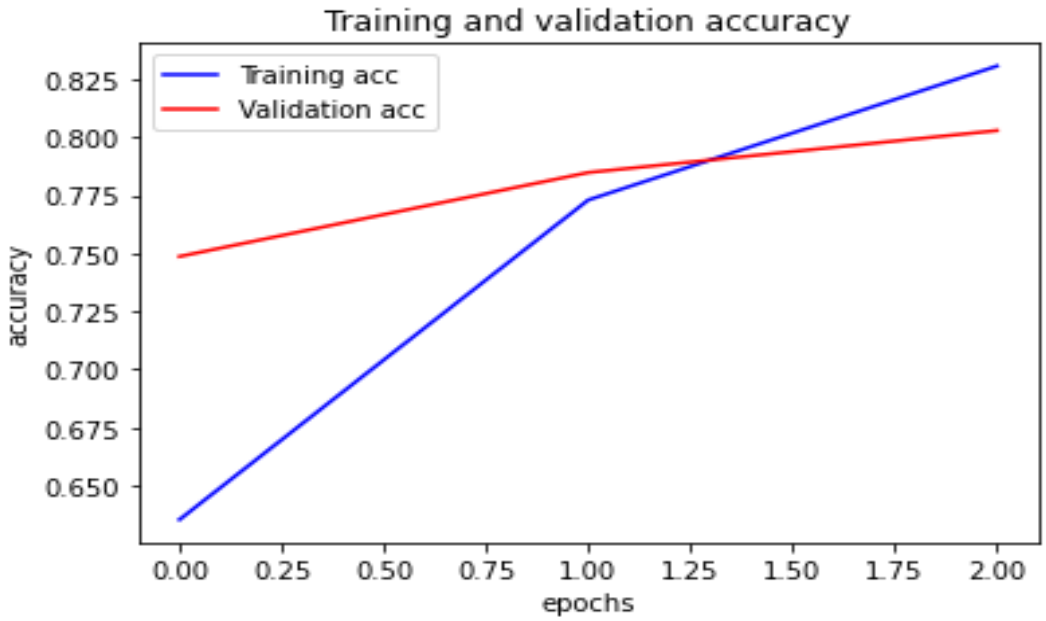
```
training = model.fit(train_x, train_y, epochs=3, batch_size=32,
                     validation_data=(val_x, val_y), verbose=1)
```

```
Epoch 1/3
90/90 [==============================] - 24s 145ms/step - loss: 0.6435 - accuracy: 0.6354 - val_loss: 0.5447 - val_accuracy: 0.7486
Epoch 2/3
90/90 [==============================] - 7s 79ms/step - loss: 0.4722 - accuracy: 0.7729 - val_loss: 0.4918 - val_accuracy: 0.7847
Epoch 3/3
90/90 [==============================] - 7s 78ms/step - loss: 0.3919 - accuracy: 0.8306 - val_loss: 0.4722 - val_accuracy: 0.8028
```

6.1 Model Training

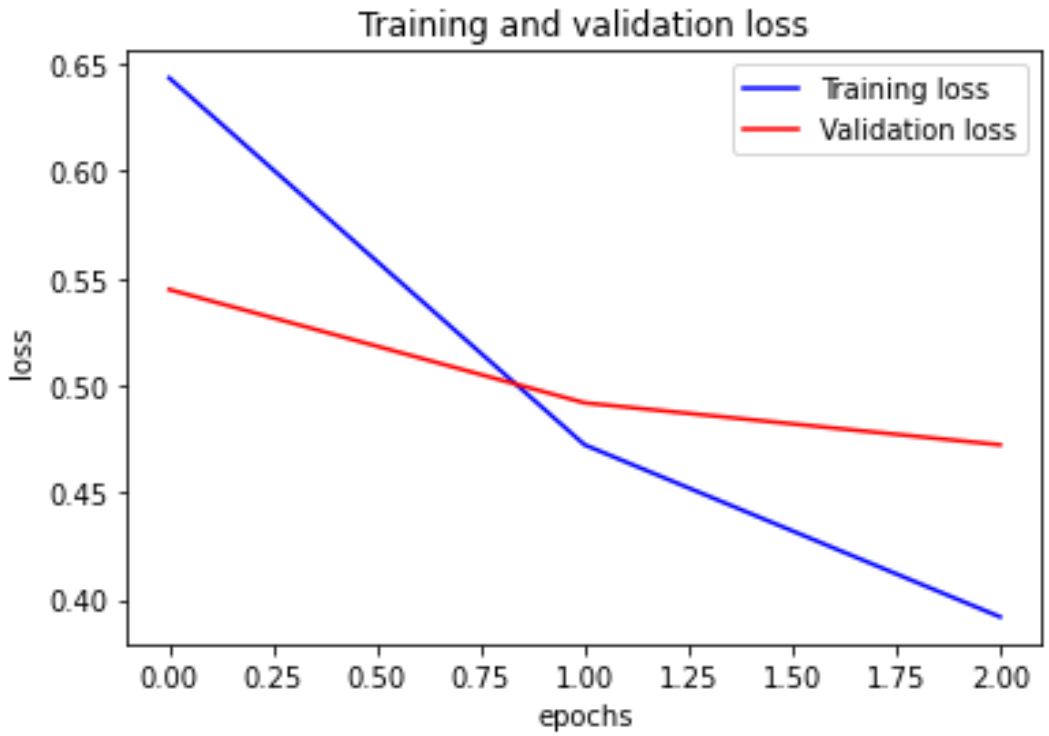
6.2 Resultant Graphs

6.2.1 Accuracy Vs Epochs Graph



6.2.1 Accuracy vs Epochs graph

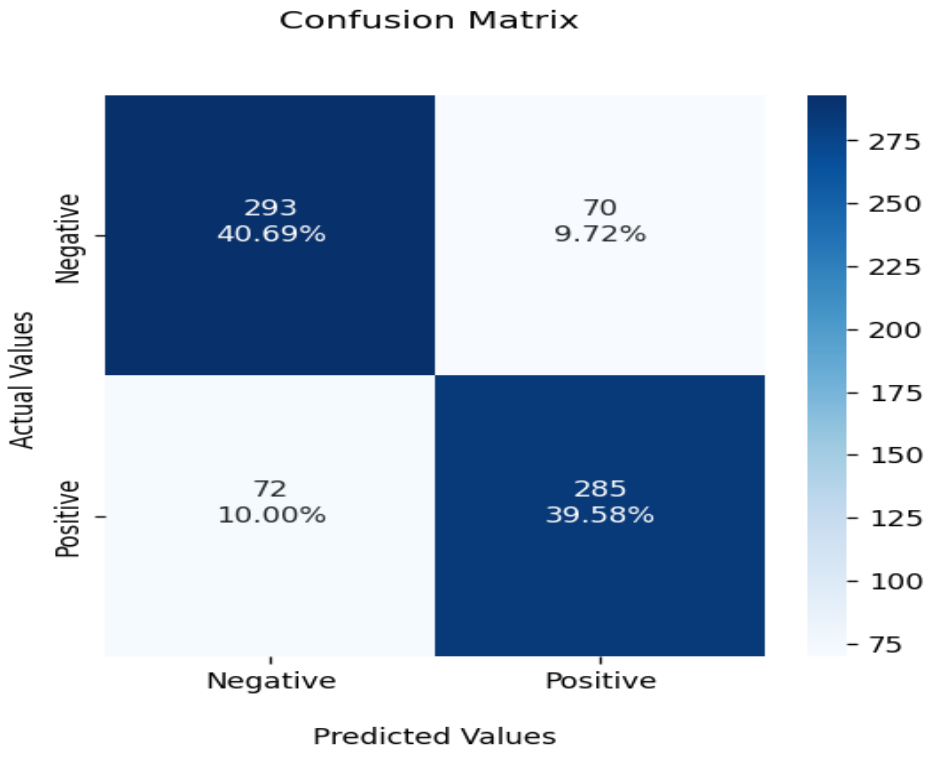
6.2.2 Loss Vs Epochs Graph



6.2.2 Loss vs Epochs graph

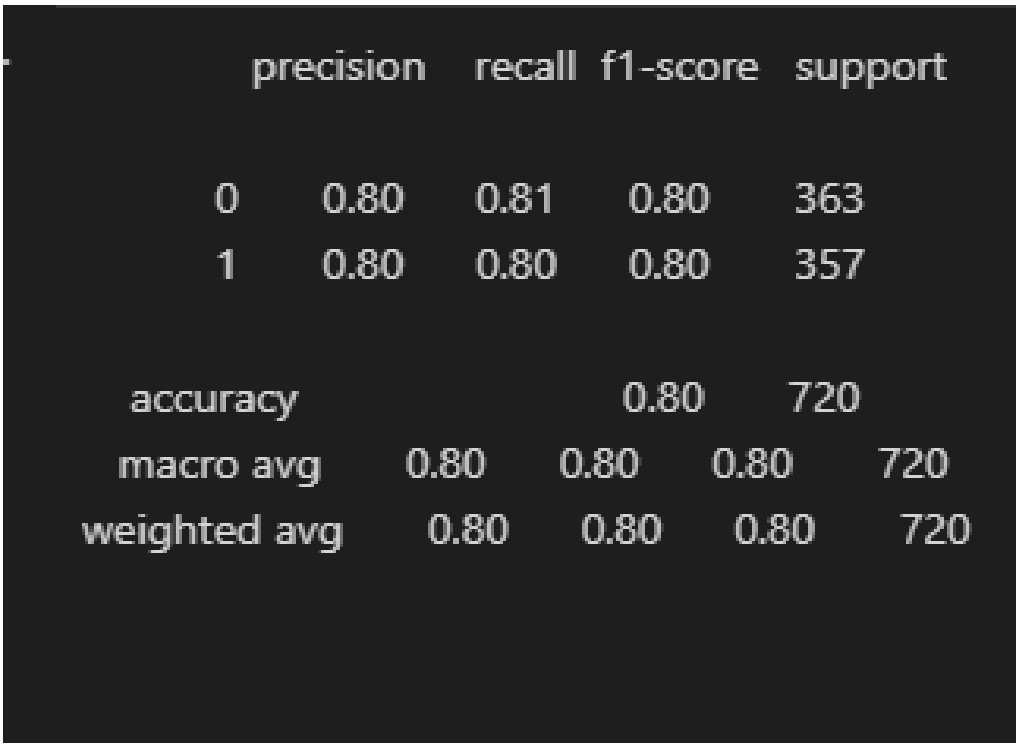
6.3 Performance Metrices

6.3.1 Confusion Matrix



6.3.1 Confusion Matrix

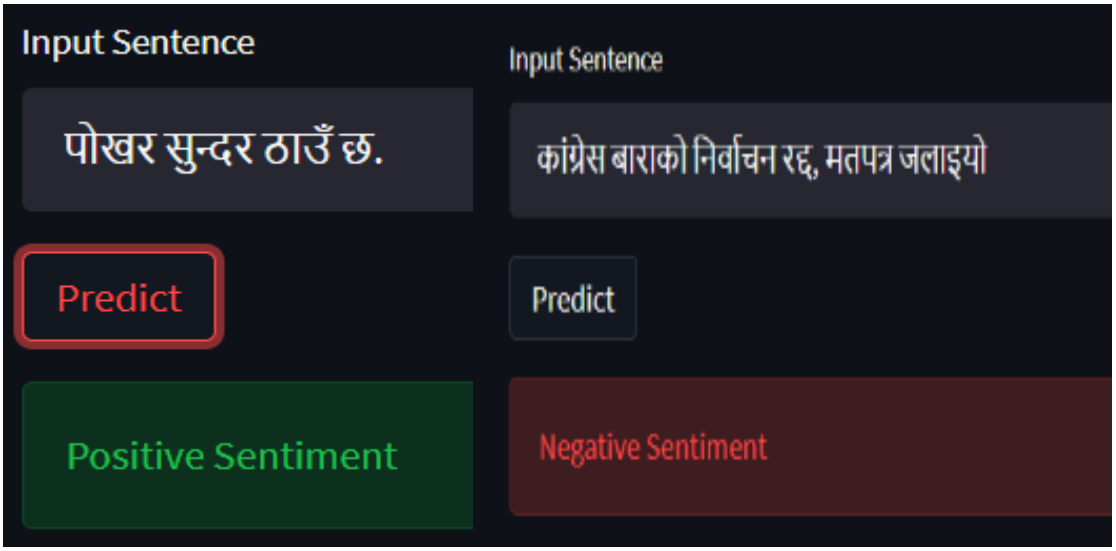
6.3.2 Precision, Recall, and F1 Score



	precision	recall	f1-score	support
0	0.80	0.81	0.80	363
1	0.80	0.80	0.80	357
accuracy			0.80	720
macro avg	0.80	0.80	0.80	720
weighted avg	0.80	0.80	0.80	720

6.3.2 Precision, Recall, and F1 Score

6.4 Prediction Result



6.4 Prediction Results

CHAPTER 7

CONCLUSION, LIMITATIONS AND FUTURE WORK

7.1 Conclusion

The sentiment classification tool made by using the neural network is necessary for various business organizations as well as governmental organizations to analyze all the comments or reviews about a product/policy and then applying changes to those policies and products. This tool will give a huge boost to these organization in analyzing the sentiment and take better actions based on the sentiment. The sentiments can now be classified as “positive” and “negative” by using our model. Thus, “Sentiment analysis using deep neural network for Nepali language” analyzes the Nepali texts for sentiment and produces the sentiment polarity making business and policies analysis easier.

7.2 Limitation

Some of the limitations of our model are:

- It gets confused with the sarcastic text.
- Sentiment Prediction on English Nepali mix up text results in false prediction.
- Ambiguity can occur, semantic analysis is difficult to address.

7.3 Future Work

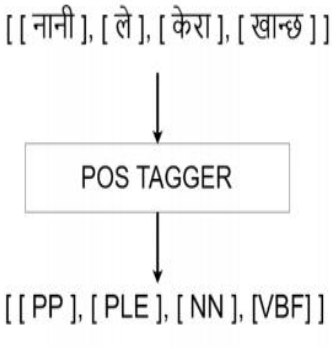
- **Pos Tagging**

  POS tagging is a process that assigns parts of speech to each word and some other token. From the word vectors and their respective POS (Parts of Speech) tags need to be placed by the system. POS tagging gives more insight to the data making neural network little bit easier to classify sentiment data. Some of the list of POS tags can be seen in the table below.

7.3 POS Tags example from NELRALEC Tagset

POS Tags	Description	Example
NN	Common Noun	केटो, केटा, कलम
NP	Proper Noun	राम, युबराज
JM	Masculine Adjective	मोटो, पातलो
JF	Feminine Adjective	दुब्ली, मोटी
VI	Infinite Verb	गर्नु, नगर्नु
…	…	...

The following figure below shows the POS Tagging system with example in Nepali Language:



7.3 POS Tagging System example for Nepali Text

- Comparing the accuracy using different models.
- Mobile application

CHAPTER 8

REFERENCES

[1] B. B. Shrestha and B. K. Bal, “Named-entity based sentiment analysis of Nepali news media texts,” in Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, 2020, pp. 114–120.

[2] A. N. Muhammad, S. Bukhori, and P. Pandunata, “Sentiment analysis of positive and negative of YouTube comments using naïve Bayes – support vector machine (NBSVM) classifier,” in 2019 International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), 2019.

[3] T. B. Shahi and A. K. Pant,“Nepali news classification using Naïve Bayes, Support Vector Machines and Neural Networks,” in 2018 International Conference on Communication information and Computing Technology (ICCICT),2018,pp.1–5.

[4] Thapa, L. B., & Bal,B. K.(2016). Classifying sentiments in Nepali subjective texts.2016 7th International conference on information, intelligence, systems & applications (IISA),(pp. 1-6).

[5] M. Tripathi, “Sentiment analysis of Nepali COVID19 tweets using NB, SVM and LSTM,” September 2021, vol. 3, no. 3, pp. 151–168, 2021.

[6] Singh, M., Jakhar, A. K., & Pandey, S. (2021). Sentiment analysis on the impact of coronavirus in social life using the BERT model. Social Network Analysis and Mining, 11(1), 1-11.

[7] Tamrakar, S., Bal, B. K., & Thapa, R. B. (2020). Aspect Based Sentiment Analysis of Nepali Text Using Support Vector Machine and Naive Bayes. Technical Journal, 2(1), 22-29.

[8] Gupta,C. P.,& Bal,B. K.(2015). Detecting Sentiment in Nepali texts: A bootstrap approach for Sentiment Analysis of texts in the Nepali language.2015 International Conference on Cognitive Computing and Information Processing (CCIP), (pp. 1-4).

[9] Shrestha, I., & Dhakal, S.S. (2016). A new stemmer for Nepali language. 2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Fall), 1-5

[10] https://github.com/oya163/nepali-sentiment-analysis(2022)