**A MINI PROJECT REPORT**

**On**

**Deep Learning based Text Classification**

*Submitted by,*

ALWAL DHANASHREE 20J41A6604

CHIPPADA SNEHA MANASA 20J41A6615

D MARK PREETHAM 20J41A6616

NAGULA CHANDANA 20J41A6637

*in partial fulfilment of the requirements for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Under the Guidance of

## Mr. B Srinath

Assistant Professor, Computer Science and Engineering (AIML)

## COMPUTER SCIENCE AND ENGINEERING (AIML)

## MALLA REDDY ENGINEERING COLLEGE

(An UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to

JNTUH, Hyderabad) Maisammaguda, Secunderabad, Telangana, India 500100

**OCTOBER- 2023**

# MALLA REDDY ENGINEERING COLLEGE

Maisammaguda, Secunderabad, Telangana, India 500100



# BONAFIDE CERTIFICATE

This is to certify that this minor project work entitled “**Deep Learning Based Text Classification**”, submitted by **ALWAL DHANASHREE (20J41A6604) , CHIPPADA SNEHA MANASA (20J41A6615) , D MARK PREETHAM (20J41A6616) , NAGULA CHANDANA (20J41A6637)** to Malla Reddy Engineering College affiliated to JNTUH, Hyderabad in partial fulfilment for the award of **Bachelor of Technology** in COMPUTER SCIENCE AND ENGINEERING is a *bonafide* record of project work carried out under my/our supervision during the academic year 2023 – 2024 and that this work has not been submitted elsewhere for a degree.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Mr B. Srinath** | **Dr. S Shiva Prasad** |
| **PROJECT GUIDE** | **HOD** |
| Assistant Professor (CSE-AIML) | CSE (AIML) |
| Malla Reddy Engineering College | Malla Reddy Engineering College |
| Secunderabad, 500 100 | Secunderabad, 500 100 |
|  |  |

**Submitted for Minor Project viva-voce examination held on \_\_\_\_\_\_\_\_\_**

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## 

## MALLA REDDY ENGINEERING COLLEGE

Maisammaguda, Secunderabad, Telangana, India 500100

**ACKNOWLEDGEMENT**

We express our sincere thanks to our **Principal, Dr. A. Ramaswami Reddy**, who took keen interest and encouraged in every effort during the project work.

We express our heartfelt thanks to **Dr. S. Shiva Prasad**, **Head and Professor,** Department of Computer Science and Engineering (AIML), for his kind attention and valuable guidance throughout the project work.

We are thankful to our Project Coordinator, **Mr. B. Srinath,** **Assistant Professor**, Department of Computer Science and Engineering, for his cooperation during the project work.

We are extremely thankful to our Project Guide **Dr. Nagendra M,** for his constant guidance and Support to complete the project work.

We also thank all the teaching and non-teaching staff of Department for their cooperation during the project work.

**ALWAL DHANASHREE 20J41A6604**

**CHIPPADA SNEHA MANASA 20J41A6615**

**D MARK PREETHAM 20J41A6616**

**NAGULA CHANDANA 20J41A6637**

### ABSTRACT

Text classification deals with unstructured / raw or unknown data and categorizes them based on similarity observed between these data. In this Project we are going to develop a model which will analyse raw data and train the model to categorize the data based on the features we can find using a Neural Network.Deep Learning is based on the concept of Artificial neural network which tries to replicate on how neurons work in our brain. In this we have multiple steps which processes data to progressively extract higher level features from the data.There are various Neural Network Mechanisms or models which have been used for Text Classification such as CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory) Models.Text Classification was previously done with Machine Learning algorithms like Logistic Regression, Random Forest and K-Nearest Neighbours but were outperformed with various Deep Learning Algorithms.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
|  | **ABSTRACT** | **Iv** |
|  | **LIST OF FIGURES** | **Viii** |
|  | **ABBREVIATIONS** | **ix** |
| **CHAPTER** | **DESCRIPTION** | **PAGE NO.** |
| **1** | **INTRODUCTION** | **1-2** |
| **2** | **SYSTEM ANALYSIS**  2.1 EXISTING SYSTEM  2.2 EXISTING SYSTEM DISADVANTAGES  2.3 PROPOSED SYSTEM  2.4 PROPOSED SYSTEM ADVANTAGES | **3-6**  3  4  4  5 |
| **3** | **SYSTEM STUDY** | **6-11** |
|  | 3.1 FEASIBILITY STUDY | 6 |
|  | 3.1.1 ECONOMICAL FEASIBILITY | 6 |
|  | 3.1.2 TECHNICAL FEASIBILITY | 7 |
|  | 3.1.3 SOCIAL FEASIBILITY  3.2 PRELIMINARY INVESTIGATION  3.2.1 REQUEST CLARIFICATION  3.2.2 FEASIBILITY ANALYSIS  3.2.3 REQUEST APPROVAL  3.3 SYSTEM DESIGN AND DEVELOPMENT | 7  7  8  8  9  9 |
|  | 3.3.1 INPUT DESIGN | 9 |
|  | 3.3.2 OUTPUT DESIGN | 10 |
| **4** | **SYSTEM DESIGN** | **11-12** |
|  | 4.1 SYSTEM ARCHITECTURE | 11 |
|  | 4.2 DATA FLOW DIAGRAM | 12 |
| **5** | **FLOW CHARTS ANDSEQUENCE DIAGRAM** | **13-15** |
|  | 5.1 FLOW CHARTS | 13 |
|  | 5.1.1 ENCODER | 13 |
|  | 5.1.2 DECODER | 14 |
|  | 5.2 SEQUENCE DIAGRAM | 15 |
|  |
|  |
| **6** | **IMPLEMENTATION** | **16** |
|  | 6.1 MODLUES | 16 |
|  | 6.1.1 TENSORFLOW | 16 |
|  | 6.1.2 MATPLOTLIB | 16 |
|  | 6.1.3 NUMPY | 16 |
| **7** | **SYSTEM REQUIREMENTS** | **17** |
|  | 7.1 HARDWARE REQUIREMENTS | 17 |
|  | 7.2 SOFTWARE REQUIREMENTS | 17 |
|  | 7.3 SOFTWARE DESCRIPTION | 18 |
|  | 7.4 PYTHON FEATURES | 20 |
|  | 7.4.1 TENSORFLOW  7.4.2 NUMPY  7.4.3 MATPLOTLIB | 21  21  22 |
|  |
| **8** | **SYSTEM TESTING** | **22- 26** |
|  | 8.1 TESTING METHODOLOGIES | 22 |
|  | 8.1.1 UNIT TESTING | 23 |
|  | 8.1.2 INTEGRATION TESTING | 23 |
|  | 8.1.3 USER ACCEPTANCE TESTING | 23 |
|  | 8.1.4 OUTPUT TESTING | 24 |
|  | 8.1.5 VALIDATION CHECKING | 24 |
|  | 8.2 USER TRAINING | 25 |
|  | 8.3 MAINTANANCE | 26 |
| **9** | **SCREENSHOTS** | **27 - 33** |
|  | 9.1 MODEL STRUCTURE  9.2 DATASET  9.2.1 CSV FILE  9.2.2 DATAFRAME  9.2.3 PLOTS | 27  28  29  29  29 |
|  |
|  |
|  |
|  | 9.3.1 MODEL PERFORMANCE  9.3.2 CALCULATION METRICS  9.4 PLOTS | 31  32  33 |
| **10** | **CONCLUSION** | **34** |
|  | **REFERENCES** | **35-36** |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO.** | **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 1. | 4.1 | MODEL ARCHITECTURE | **11** |
| 2. | 4.2 | DATA FLOW DIAGRAM | **12** |
| 3. | 5.1 | ENCODER | **13** |
| 4. | 5.2 | DECODER | **14** |
| 5. | 5.3 | SEQUENCE DIAGRAM | **15** |
| 6. | 9.1.1 | MODEL STRUCTURE -1 | **27** |
| 7. | 9.1.2 | MODEL STRUCTURE - 2 | **27** |
| 8. | 9.1.3 | MODEL STRUCTURE - 3 | **28** |
| 9. | 9.4 | CSV FILE | **28** |
| 10. | 9.5 | DATAFRAME | **29** |
| 11. | 9.6 | PIE CHART - DATA | **30** |
| 12. | 9.7 | LINE PLOT | **31** |
| 13. | 9.8 | MODEL PERFORMANCE | **32** |
| 14. | 9.9 | CONFUSION MATRIX | **33** |
| 15. | 9.10 | METRICS | **34** |
| 16. | 9.11 | PIE CHART | **34** |
|  |  |  |  |

**ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **SNO:** | **ABBREVIATIONS** | **FULLFORM** |
| 1. | ML | MACHINE LEARNING |
| 2. | BERT | BIDIRECTIONAL ENCODER REPRESENTATIONS FOR TRANSFORMERS |
| 3. | GNU | GNU’S NOT UNIX |
| 4. | GPL | GENERAL PUBLIC LICENSE |
| 5. | PHP | HYPERTEXT PREPROCESSING |
| 6. | PERL | PRACTICAL EXTRACTION AND REPORTING LANGUAGE |
| 7. | WWW | WORLD WIDE WEB |
| 8. | OOP | OBJECT ORIENTED PROGRAMMING |

**CHAPTER 1**

#### INTRODUCTION

Why do we need Text Classification?

Customers and buyers often check on reviews before purchasing a product and analyse on how good a product is by factors of price, quantity purchased, stars and reviews placed on them. Although it is easy to read and classify features such as price and quantity, analysing text is different. Humans are cognizant of the knowledge they have adapted to a specific language, they are able to mostly understand the hidden intent, sarcasm or other critical aspects of speech notions which is not present in machines. Machine Learning (ML) find it hard to decipher the true intent and meaning of a sentence due to its inherent complexity to understand something which is not in a binary form.

Text classification can be described as a machine learning technique to classify the type of text into a particular category. These categories differ based on task they perform such as sentiment analysis, topic labelling, spam detection and intent detection. As data that available in the natural world is mostly unstructured, it becomes crucial to find a way to manage these datasets to form a desirable structure to analyse these problems.

Purpose Text classification is a widely accepted and adopted technique in organizations to mine and analyse unstructured and semi-structured data. With advancement of technological computing, deep learning has become more popular among academicians and professionals to perform mining and analytical operations. In this work, the authors study the research carried out in field of text classification using deep learning techniques to identify gaps and opportunities for doing research. Design/methodology/approach.

The authors adopted bibliometric-based approach in conjunction with visualization techniques to uncover new insights and findings. The authors collected data of two decades from Scopus global database to perform this study. The authors discuss business applications of deep learning techniques for text classification. Findings The study provides overview of various publication sources in field of text classification and deep learning together. The study also presents list of prominent authors and their countries working in this field. The authors also presented list of most cited articles based on citations and country of research. Various visualization techniques such as word cloud, network diagram and thematic map were used to identify collaboration network. Originality/value.

The study performed in this paper helped to understand research gaps that is original contribution to body of literature. To best of the authors' knowledge, in-depth study in the field of text classification and deep learning has not been performed in detail. The study provides high value to scholars and professionals by providing them opportunities of research in this area.

#### CHAPTER 2

#### SYSTEM ANALYSIS

**2.1 EXISTING SYSTEM**

* Text Classification was previously done using Traditional Machine Learning Models where supervised data where categorised based on filters and were classified into various groups. This model although gave high accuracy on the structured data which was known had relatively very low accuracy in unstructured or raw data.
* This was later approached using Deep Neural Models or the Neural Networks model which mapped the hidden features between the strings or words in order to derive the relation in the form of vectors, these models had a sequential approach which took higher training time due to it ready word by word and assigning a vector

to it.

* These models also had lower accuracy due to same words which differed in meaning based on how they were arranged in a sentence. These words were associated with the same value and hence the meaning of the words were based on single sentence and hence decreased the accuracy of the model.
* In these models Sentiment Analysis Concept was introduced where it took the textual data and mapped them into vectors (numeric) and then were introduced to the neural network architecture to study the layers in between them.
* This model then tried to classify based on these features on which category the belong to with an accuracy of 75.01% and a precision of 76.02% and recall of 72.18%.

**2.2** **EXISTING SYSTEM DISADVANTAGES**

* In the existing work, the system had a higher computation and training time due to its Sequential approach
* This system is less performance due to the system deriving same values for words with different meanings.

**2.3 PROPOSED SYSTEM**

**Data Description**

IMDB dataset having 50K movie reviews for natural language processing or Text analytics.  
This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training and 25,000 for testing. So, predict the number of positive and negative reviews using either classification or deep learning algorithms.

**Student’s performance Model**

In this system the model took a transformers approach where it introduced the concept of multi-head attention where rather than mapping vectors on each word the combinations of words or sentences were mapped in this model. This allowed words which had the same characters but different meanings based on how they were written in sentences to be having different interpretations which greatly improved the accuracy of the model predictions.

Due the model having an attention mechanism it allowed parallel training where multiple sentences can be analysed at a time which is not possible in a sequential model, which greatly reduced the training time for the model and worked on huge datasets at the same time.

This mechanism then was stacked with multiple layers of encoders and decoders which performed the multi-head attention mechanism, normalisation and addition.

**Attention Mechanism**

As our aim is to investigate the textual data and map them into vectors which represent different values based on words and their order in sentences, this above process is called positional encoding which maps values based on positions of words in the sentences.

Positional encoding is introduced before sending the data through the attention mechanism which then derives the categories it belongs to by passing them through multiple encoders and decoders blocks where output of each layer is mapped as input to other layers by normalizing them and adding them at each layer.

This Attention Mechanism is one of the most popular sentiments analysing mechanism used for natural language processing approaches. It keeps repeating the data until a higher accuracy and patters between these texts are properly analysed.

**2.4 PROPOSED SYSTEM ADVANTAGES**

* The model classifies the reviews with higher accuracy.
* The model can accept huge amounts of data and computes them in lesser training interval.

**CHAPTER 3**

**SYSTEM STUDY**

**3.1 FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**3.1.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**3.1.2 TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**3.1.3 SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**3.2 PRELIMINARY INVESTIGATION**

The first and foremost strategy for development of a project starts from the thought of designing a model which enables the individual, group or organization to actively classify unknown data into set of defined categories based on the similarities observed between them. When it is approved by the organization and our project guide the first activity, i.e., preliminary investigation begins. The activity has three parts:

* **Request Clarification**
* **Feasibility Study**
* **Request Approval**

**3.2.1 REQUEST CLARIFICATION**

After the approval of the request to the organization and project guide, with an investigation being considered, the project request must be examined to determine precisely what the system requires.

Here our project is basically meant to classify data with increasing complexity of data which is generated every second. This model aims on classifying reviews in order to analyze customer behaviorand is used to distinguish between spam emails and calls from any malicious attempts in damaging or accessing private information.

**3.2.2 FEASIBILITY ANALYSIS**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analysed are

* **Operational Feasibility**
* **Economic Feasibility**
* **Technical Feasibility**

**Operational Feasibility**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

**Economic Feasibility**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a based on the development environment and the software components and libraries were already available and free to use thus the cost on project of software was also low.

**Technical Feasibility**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs the development environment for python and libraries versions which are not available in the current existing versions of the application, hence requires them to manually be installed via custom environments or installed via GitHub. The system is technically feasible for development and can be developed with the existing facility.

**3.2.3 REQUEST APPROVAL**

Not all request projects are desirable or feasible. Some organization receives so many project requests from client users that only few of them are pursued. However, those projects that are both feasible and desirable should be put into schedule. After a project request is approved, it cost, priority, completion time and personnel requirement is estimated and used to determine where to add it to any project list. Truly speaking, the approval of those above factors, development works can be launched.

**3.3 SYSTEM DESIGN AND DEVELOPMENT**

**3.3.1 INPUT DESIGN**

Input Design plays a vital role in the development of the model , the library versions should have all the existing modules and the dataset must be properly cleaned without any null or NaN values which denote failed occurrences in our dataset and hence needs to be properly observed and changed as per the requirements of the model.

This system has input screens in almost all the modules. Error messages are developed to alert the user whenever he commits some mistakes and guides him in the right way so that invalid entries are not made. Let us see deeply about this under module design.

Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors. The error is in the input are controlled by the input design. The application has been developed in user-friendly manner. The model input has been designed in a manner where it takes a set collection of data which then categories them for each and every data element present in each of that collection and retrieves the efficiency of the model.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

3.3.2 OUTPUT DESIGN

The Output from the computer is required to mainly depict the categories of the data and which data items, in this case the reviews which are determined to be positive or negative are then analysed and differentiated to their categories. The model shows us the reviews which are observed to be positive and show positive feedback on the movies and the reviews which are observed to be negative show negative feedback on the movies.

The application starts running when it is executed for the first time. The model requires the data to be sampled to numeric data instead of text in order to process it properly which is already done in the dataset library and requires enough computational time for the model to learn and derive output based on the quantity of reviews which have to analysed.

#### CHAPTER 4

#### SYSTEM DESIGN

**4.1 SYSTEM ARCHITECTURE**

**Architecture Diagram**

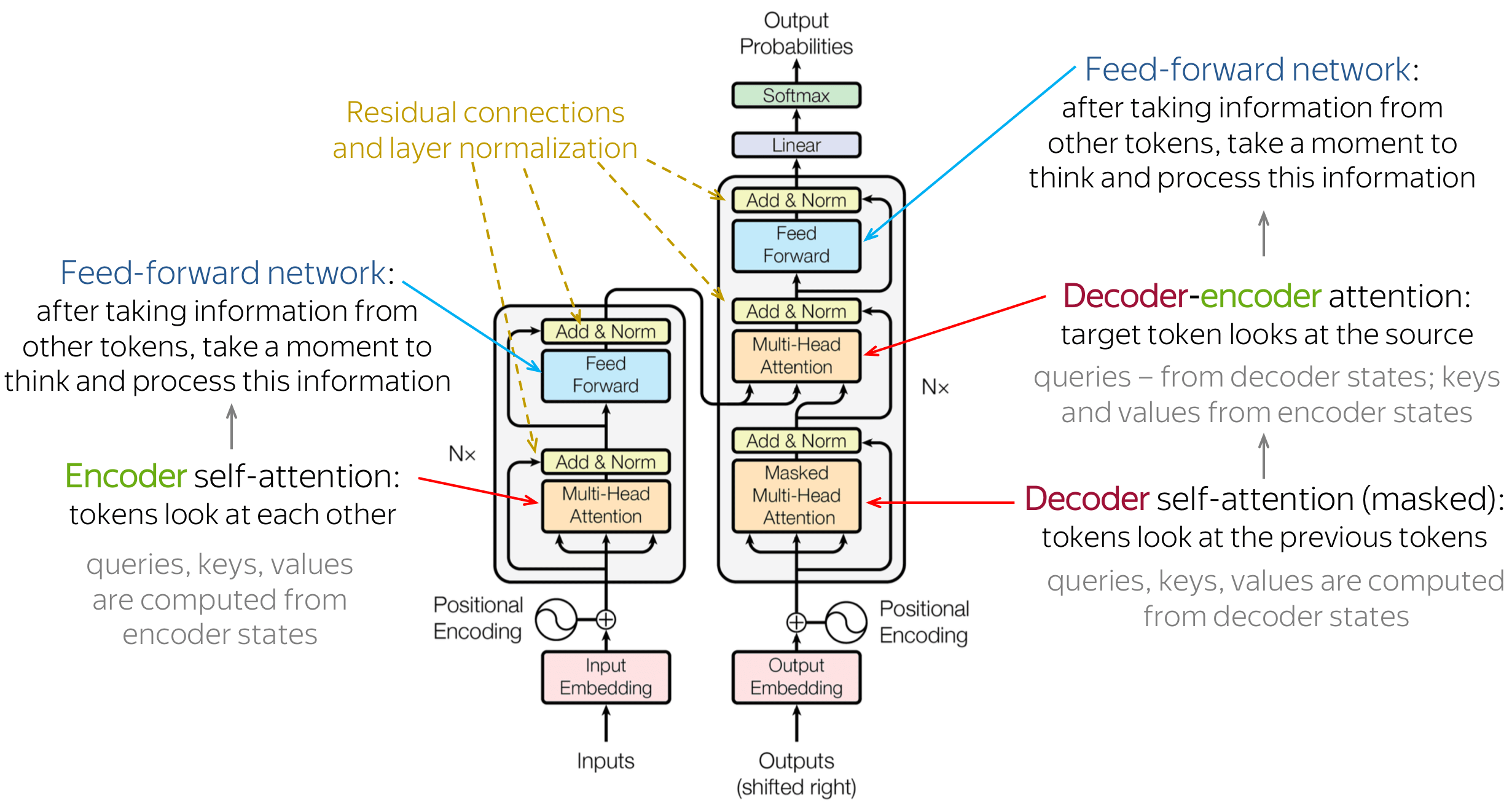


Fig 4.1

**4.2 DATA FLOW DIAGRAM**

Response

Request

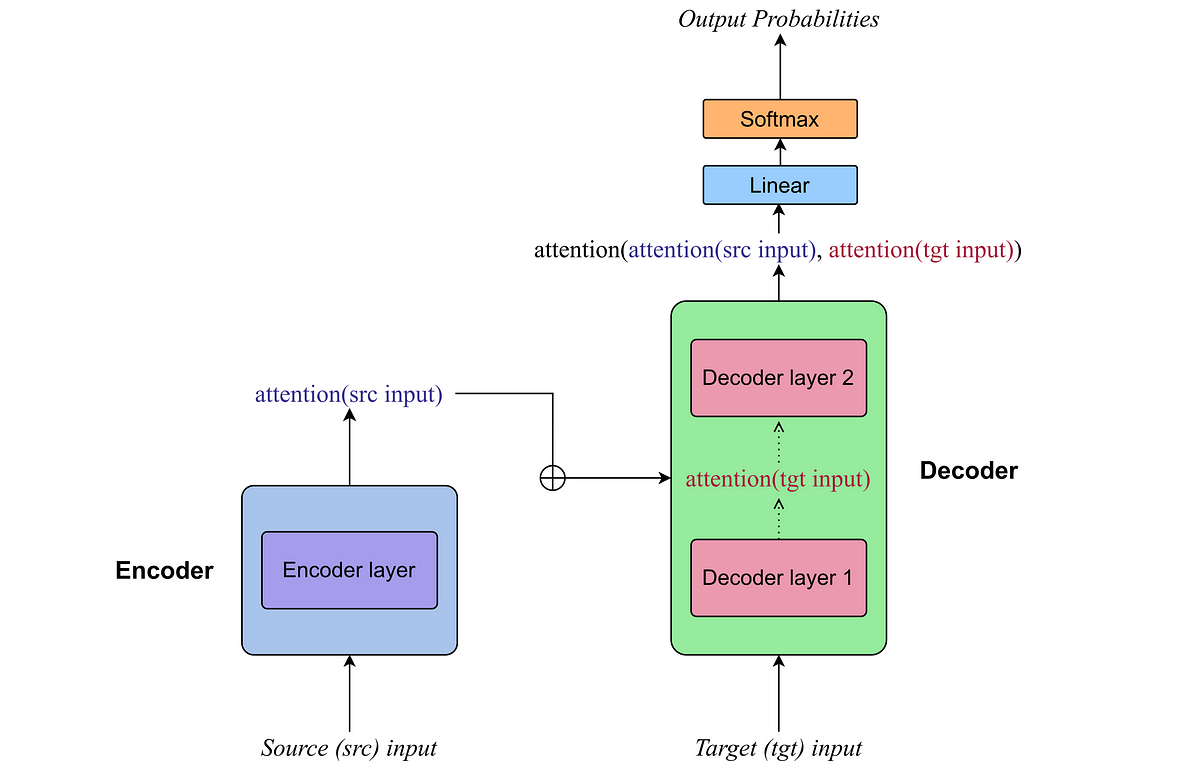


Fig 4.2

**CHAPTER 5**

**FLOW CHARTS AND SEQUENCE DIAGRAMS**

**5.1 FLOW CHARTS**

**ENCODER**

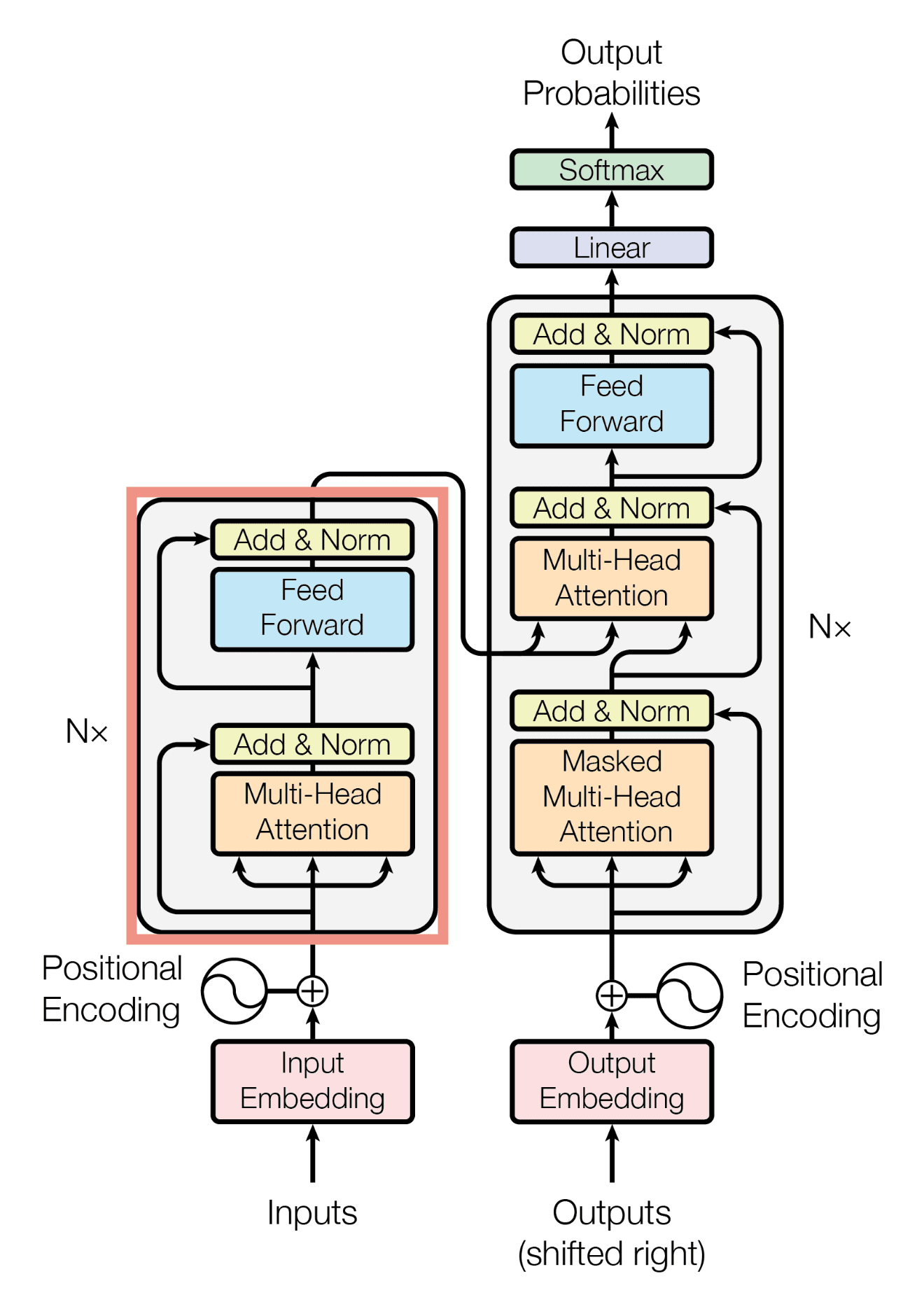


Fig 5.1

**DECODER**

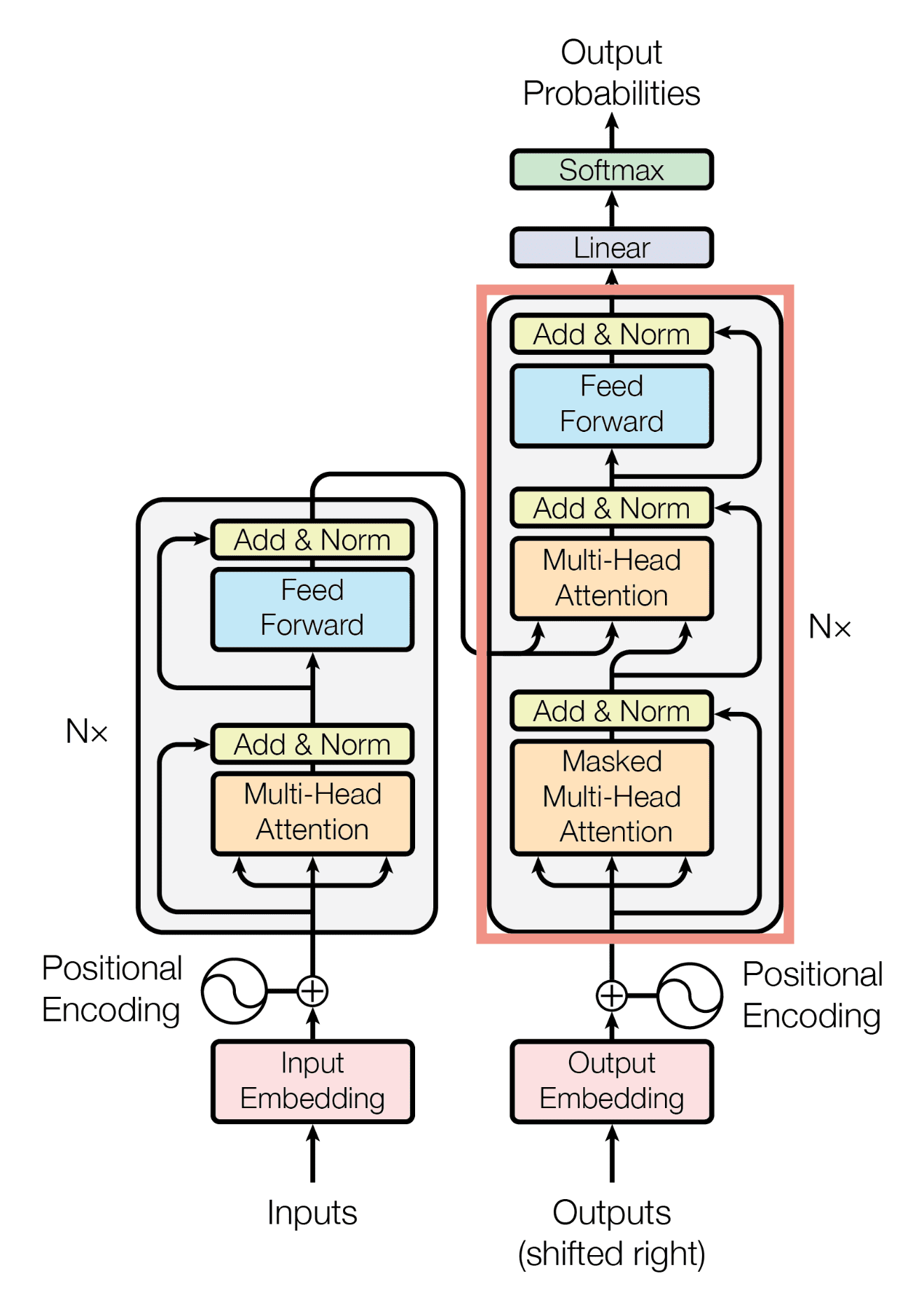


Fig 5.2

**Sequence Diagram**

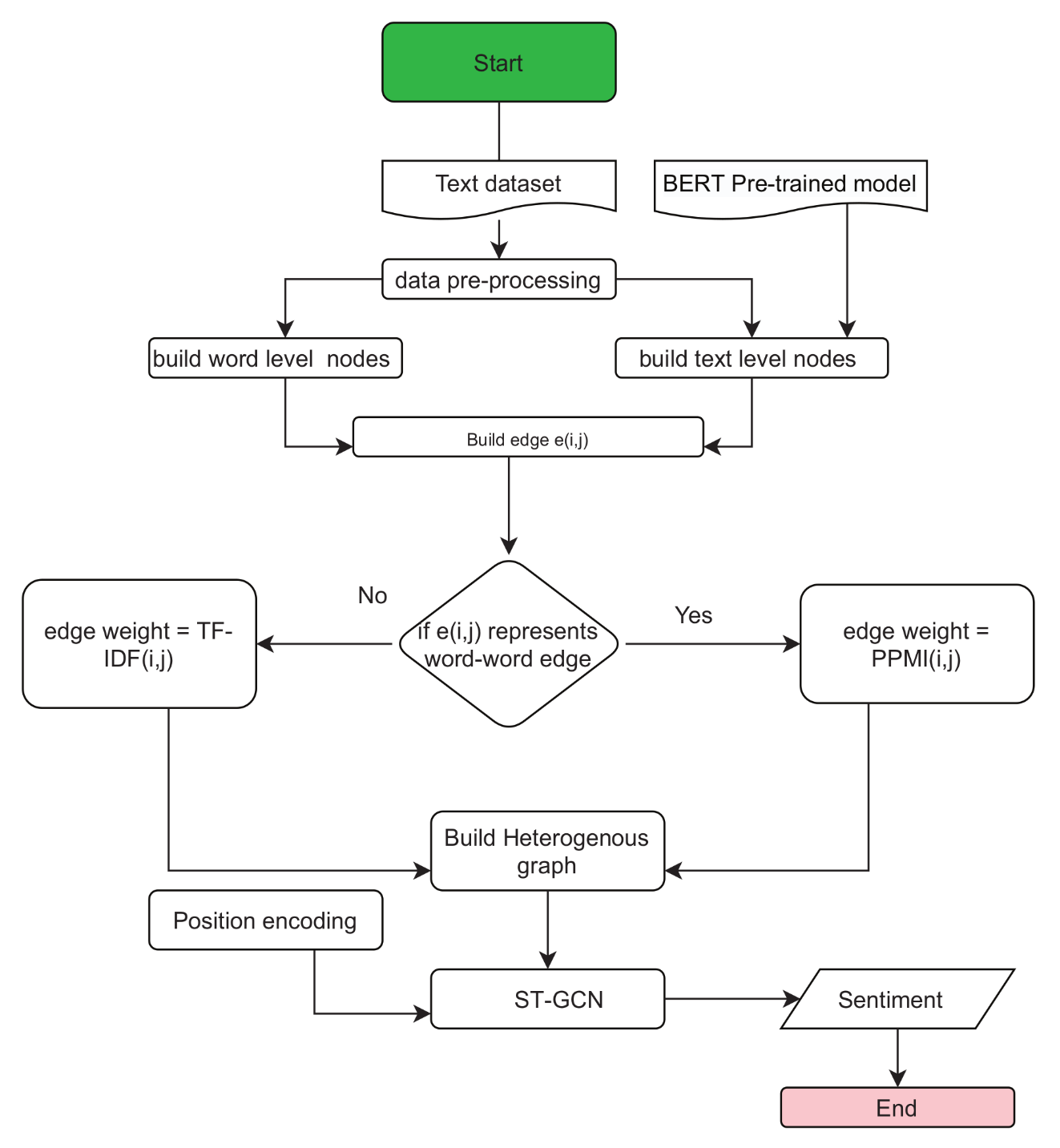


Fig 5.3

**CHAPTER 6**

#### IMPLEMENTATION

##### 6.1 MODULES

**6.1.1 Tensorflow**

In this module, the Tensorflow library which is used to build the model defines on the structure of each layer and all the functions that are performed on each layer of the Model. They also contain the attention mechanism and functions which are used in the Model, which does not require the user to implement their structure or working but can directly use them. This module is responsible in giving the model the architecture to perform.

**6.1.2 Matplotlib**

In this module, the Matplotlib is responsible for plotting and analysing the used dataset alongside visualising the model outputs and metrics to have a contrast on how the model has performed in comparison to the previous models.

**6.1.3 NumPy**

In this module, all the mathematical functions which include the normalization of values, usage of activation functions to map from various layers including input layers, hidden layers and output layers. It defines on how all the key functions of the model are implemented on each data set value.

.

**CHAPTER 7**

#### SYSTEM REQUIREMENTS

**7.1 HARDWARE REQUIREMENTS:**

* **Processor** - i3 core or above
* **RAM** - 4 GB. (min)
* **Hard Disk** - 40 GB.

**7.2 SOFTWARE REQUIREMENTS:**

* **Operating system :** Windows 8 or above.
* **Coding Language :** Python.
* **Libraries Used:**
  + Tensorflow
  + NumPy
  + Warnings
  + Matplotlib

**7.3 SOFTWARE DESCRIPTION**

**Python**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). This tutorial gives enough understanding on Python programming language.

Python is a popular programming language. It was created in 1991 by Guido van Rossum.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* System scripting.

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-orientated way or a functional way.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

**History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## 7.4 Python Features

## Python's features include −

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

**Scalable** − Python provides a better structure and support for large programs then shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X.

### Python Syntax compared to other programming languages

* Python was designed to for readability, and has some similarities to the English language with influence from mathematics.
* Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
* Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

**7.4.1 Tensorflow**

TensorFlow is an end-to-end open-source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models. See the [TensorFlow documentation](https://tensorflow.org/) for complete details on the broader TensorFlow system.

TensorFlow APIs are arranged hierarchically, with the high-level APIs built on the low-level APIs. Machine learning researchers use the low-level APIs to create and explore new machine learning algorithms. In this class, you will use a high-level API named tf.keras to define and train machine learning models and to make predictions. tf.keras is the TensorFlow variant of the open-source [Keras](https://keras.io/) API.

**7.4.2 NumPy**

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package used for the purpose of scientific computing with Python.  
Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.

**7.4.3 Matplotlib**

Matplotlib is an amazing visualization library in **Python**for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

#### CHAPTER 8

#### SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 8.1 TESTING METHODOLOGIES

The following are the Testing Methodologies:

* **Unit Testing.**
* **Integration Testing.**
* **User Acceptance Testing.**
* **Output Testing.**
* **Validation Testing**
* **System Testing**

**8.1.1 Unit Testing**

Unit testing focuses verification effort on the smallest unit of Model’s design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All-important processing path are tested for the expected results. All error handling paths are also tested.

**8.1.2 Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated

1. **Top-Down Integration**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the model is tested from main module and individual stubs are replaced when the test proceeds downwards.

**8.1.4** **OUTPUT TESTING**

After performing the validation testing, the next step is output testing of the proposed system, since no model could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

**8.1.5** **VALIDATION CHECKING**

Validation checks are performed on the following fields.

**Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single module. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested.

A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the model.

**Preparation of Test Data**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the model testing. After preparing the test data the model under study is tested using that test data. While testing the model by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

**Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a model is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the analyst’s person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

**Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

**8.2 USER TRAINING**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

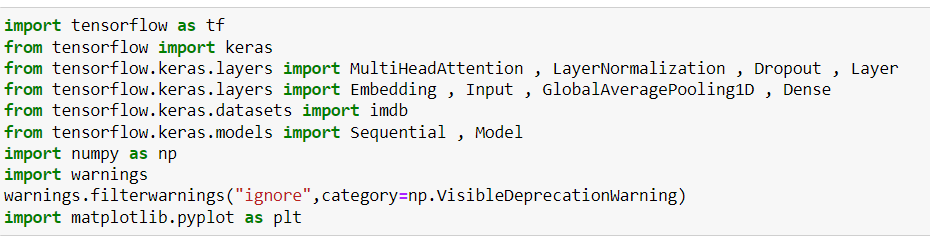
**8.3 MAINTAINENCE**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of model development. Depending on the requirements, this model has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easy.

**CHAPTER 9**

**SCREENSHOTS**

**9.1 MODEL STRUCTURE**

****



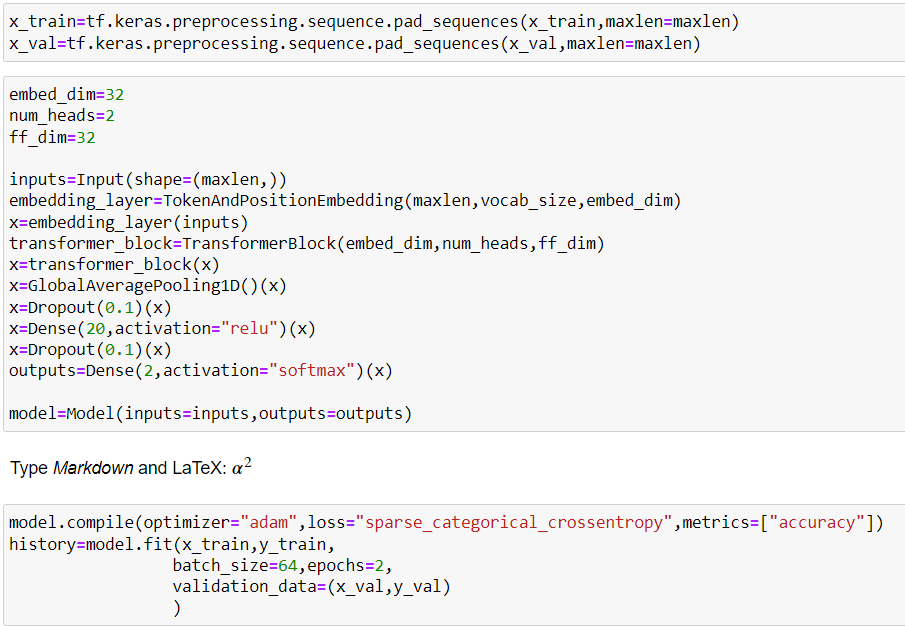


Fig 9.1, 9.2, 9.3

**9.2.1 DATA SET**

**CSV FILE:**

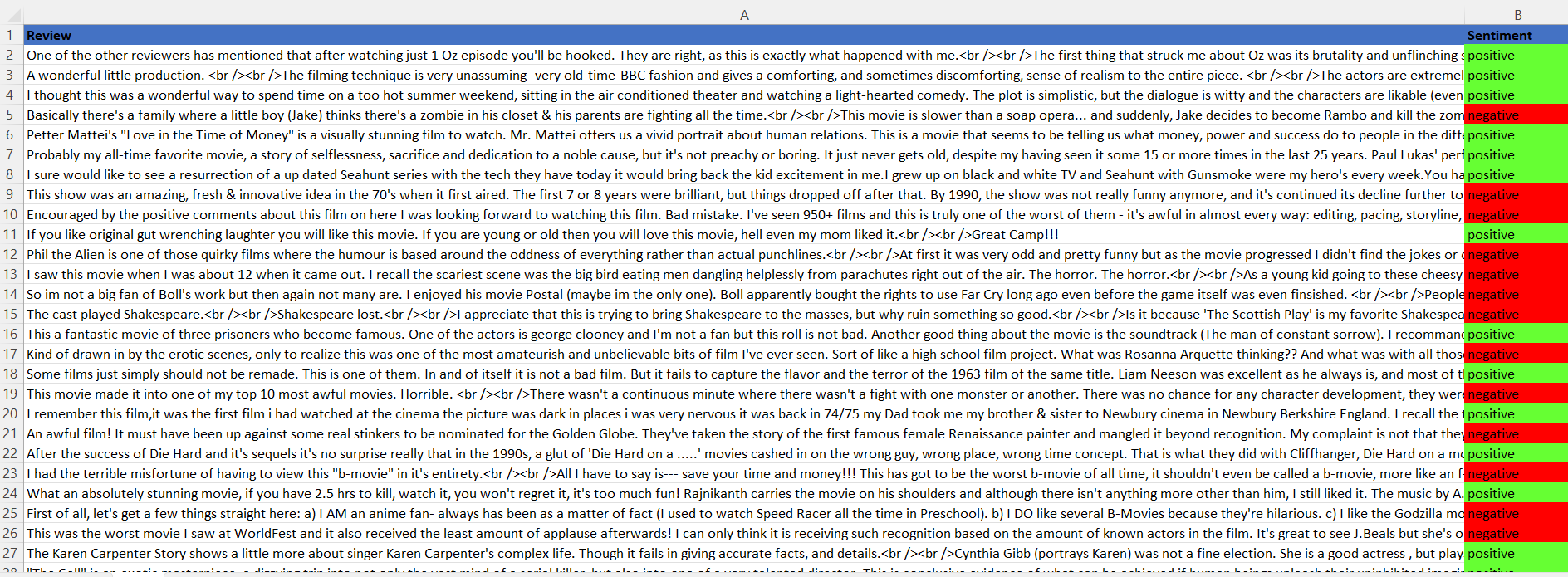
****

Fig 9.4

**DATAFRAME**

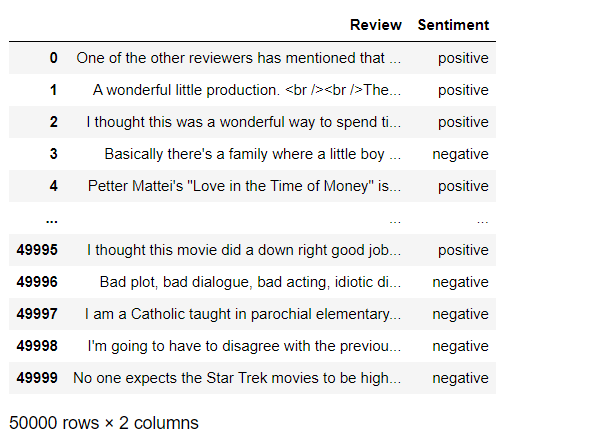


Fig 9.5

**PLOTS**

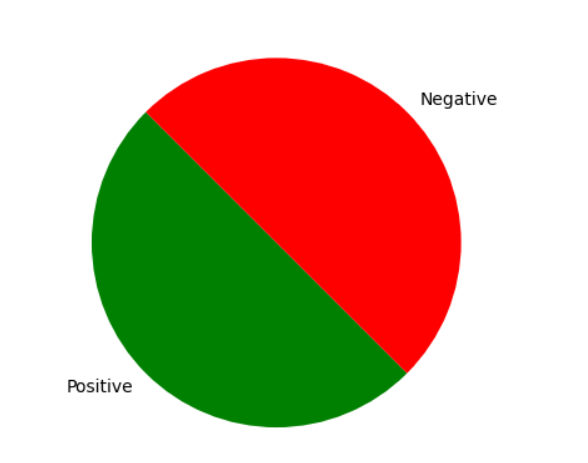
****

Fig 9.6

**LINE PLOT**

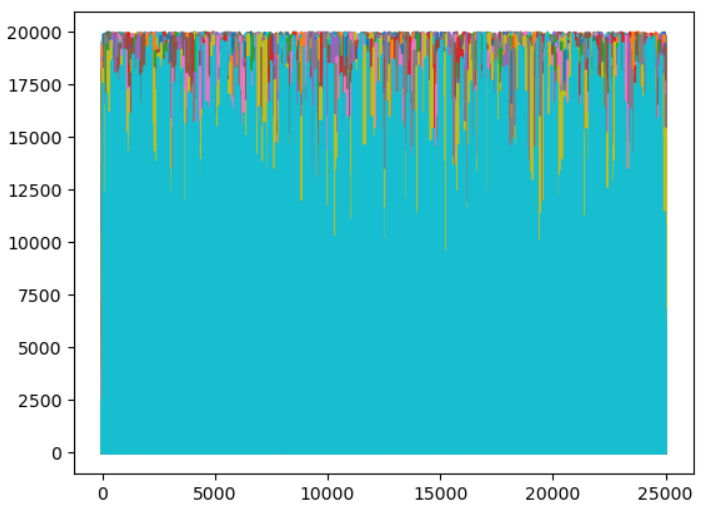
****

Fig 9.7

**9.3.1 MODEL PERFORMANCE**

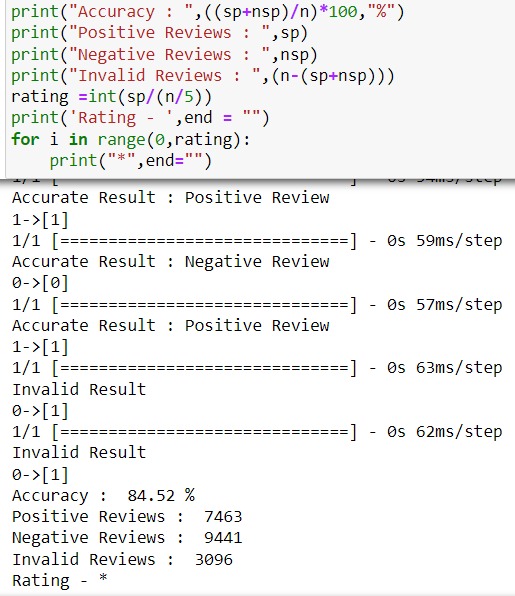


Fig 9.8

**9.1.3 CALCULATION METRICS**

**9.1.3.1 CONFUSION MATRIX**

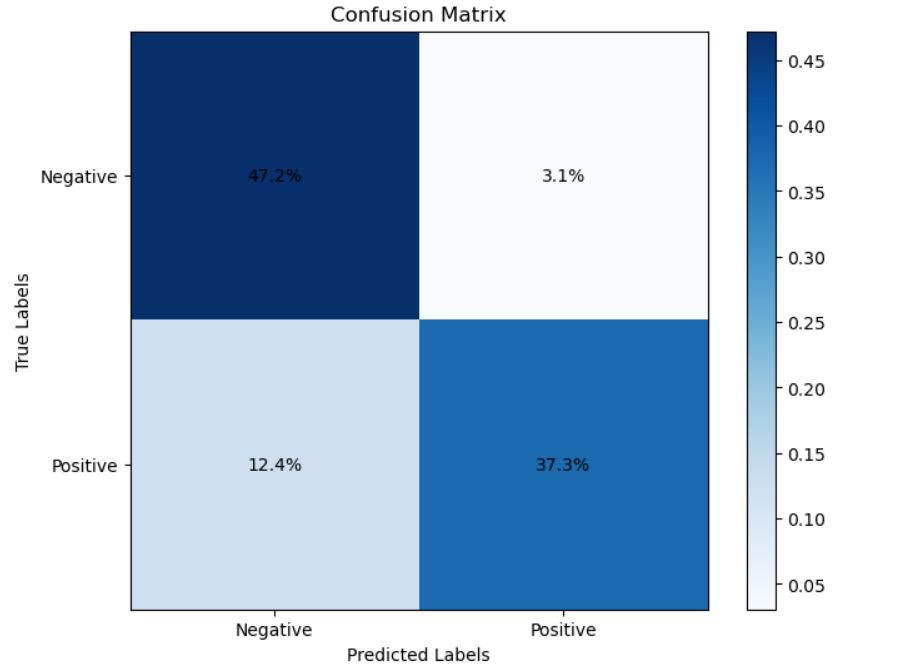
****

Fig 9.9

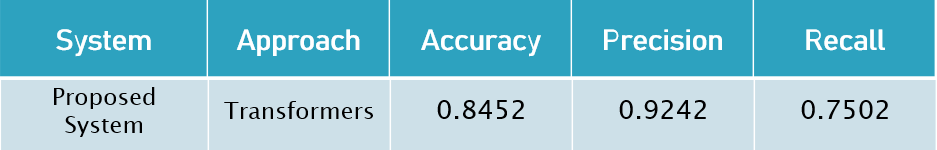
****

Fig 9.10

**9.4 PLOTS**

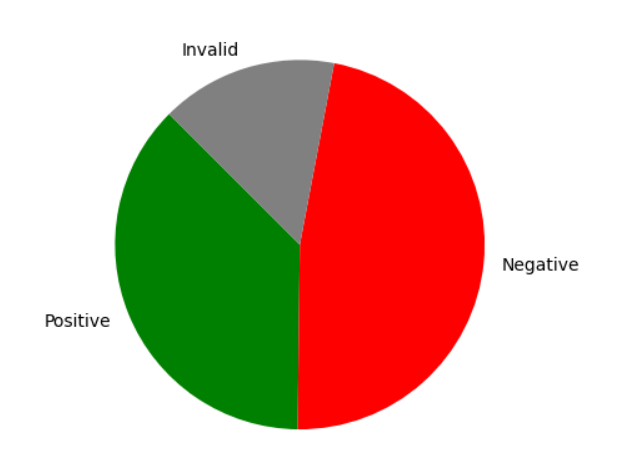
****

Fig 9.11

#### CHAPTER 10

#### CONCLUSION

The observations which we have derived from our model outperforms the other traditional and existing models with a higher accuracy than the previous models and works on huge datasets with very less training time which makes it faster and more efficient model than other modern text classification techniques.

Implementation of different transformer models can further be done to enhance the model in better converting complex data in a much faster and easier manner.  
This model can also be adapted to enforce better spam mail detection and in reviewing various categorical data and classify them.

#### REFERENCES

[1] Bharath K, Published in the year 2021 ,Transformers For Text Classification.

[2]Dhaval Patel, Published on August 29, 2021 , Text Classification Using BERT & Tensorflow.

[3]Shervin Minaee , Nal Kalchbrenner , Erik Cambria , Narjes Nikzad , Meysam Chenaghlu , Jianfeng Gao, Published on January 1st, 2020, Text Classification using Deep Learning: A Comprehensive Review

[4] Aidan Gomez, Llion Jones, Lukasz Kaiser, Niki Parmar, Illia Polosukhin, Noam Shazeer, Jakob Uszkoreit and Ashish Vaswani, published in the year 2017, Attention is All You Need - Research Article on transformers architecture

[5] Kyunghyun Cho, Bartvan Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and. Learning phrase representations using rnn encoder- decoder for statistical machine translation. CoRR,abs/1406.1078,2014.

[6] Francois Chollet. Xception: Deep learning with depth wise separable convolutions. arXiv preprintarXiv:1610.02357,2016.

[7] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modelling. CoRR, abs/1412.3555,2014.

[8] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. arXivpreprintarXiv:1705.03122v2,2017.

[9] Alex Graves. Generating sequences with recurrent neural networks. arXivpreprint arXiv:1308.0850,2013.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages770–778,2016.

[11] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies,2001.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780,1997.

[13] Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu.

Exploring the limits of language modelling. arXivpreprintarXiv:1602.02410,2016.

[14] Łukasz Kaiser and Ilya Sutskever. Neural GPUs learn algorithms. In International Conference on Learning Representations (ICLR),2016.

[15] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaronvanden Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in lineartime. arXivpreprintarXiv:1610.10099v2, 2017.

[16] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured attention networks. In International Conference on Learning Representations,2017.

[17] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[18] Oleksii Kuchaiev and Boris Ginsburg. Factorization tricks for LSTM networks. arXivpreprint arXiv:1703.10722,2017.

[19] Zhouhan Lin, Minwei Feng, Cicero Nogueirados Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. arXivpreprint arXiv:1703.03130,2017.

[20] Samy Bengio Łukasz Kaiser. Canactive memory replace attention? In Advances in Neural Information Processing Systems, (NIPS),2016