SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

Predicting Next Word in the sentence using BERT Algorithm

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE IN THE PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE

BACHELOR OF ENGINEERING
(Computer Engineering)(SEM-I)

SUBMITTED BY

Group ID : B-8

Mr. Shaikh Zaid Mohammed Mateen	Roll No. : 4251
Mr. Pathan MohammadAayan Samirkhan	Roll No. : 4221
Mr. Pathan Muaazkhan Naimkhan	Roll No. : 4222
Mr. Waghmare Avinash Bhagwat	Roll No. : 4275

Under The Guidance of

Prof. A. N. Nawathe

DEPARTMENT OF COMPUTER ENGINEERING
Amrutvahini College of Engineering, Sangamner
Amrutnagar, Ghulewadi - 422608

2022-23

AMRUTVAHINI COLLEGE OF ENGINEERING,SANGAMNER
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that,

Group ID: B-8

Mr. Shaikh Zaid Mohammed Mateen	Roll No. : 4251
Mr. Pathan MohammadAayan Samirkhan	Roll No. : 4221
Mr. Pathan Muaazkhan Naimkhan	Roll No. : 4222
Mr. Waghmare Avinash Bhagwat	Roll No. : 4275

are bonafide students of this institute and the work entitled **“Predicting Next Word in the sentence using BERT Algorithm”** has been carried out by them under the supervision of Prof. A. N. Nawathe and is approved for the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Bachelor of Engineering (Computer Engineering).

Prof. A. N. Nawathe
Internal Guide
Dept. of Computer Engg.

Dr. R. G. Tambe / Dr. D. R. Patil
Project Coordinator
Dept. of Computer Engg.

Prof. R. L. Paikrao
H.O.D.
Dept. of Computer Engg.

Dr. M. A. Venkatesh
Principal
AVCOE Sangamner

Savitribai Phule Pune University

CERTIFICATE

This is to Certify that

Mr. Shaikh Zaid Mohammed Mateen	Roll No. : 4251
Mr. Pathan MohammadAayan Samirkhan	Roll No. : 4221
Mr. Pathan Muaazkhan Naimkhan	Roll No. : 4222
Mr. Waghmare Avinash Bhagwat	Roll No. : 4275

of B.E. in Computer Engineering was examined in the

Project Examination entitled

“Predicting Next Word in the sentence using BERT Algorithm”

on . . ./. . . /2022

At

Department of Computer Engineering

Amrutvahini College of Engineering, Sangamner

. . . .

Internal Examiner   *External Examiner*

Acknowledgment

Achievement is Finding out what you have been doing and what you have to do. The higher is submit,the harder is climb. The goal was fixed and we began with the determined resolved and put in a ceaseless sustained hard work. Greater the challenge, greater was our determination and it guided us to overcome all difficulties. It has been rightly said that we are built on the shoulders of others. For everything we have achieved, the credit goes to who had really help us for project and for the timely guidance and infrastructure. Before we proceed any further, we would like to thank all those who have helped us in all the way through. To start we are thankful to Honorable Principal **Dr. M. A. Venkatesh** sir for his encouragement and support.I would like to take this opportunity to thank to our respected Head of Department **Prof. R. L. Paikrao** and Project Coordinator **Prof. R. G. Tambe** and **Prof. D. R. Patil** .And we also thank our guide **Prof. A. N. Nawathe** for guidance, care and support, which they offered whenever we needed.

Mr. Shaikh Zaid Mohammed Mateen
Mr. Pathan MohammadAayan Samirkhan
Mr. Pathan Muaazkhan Naimkhan
Mr. Waghmare Avinash Bhagwat

Abstract

In this busy world no one has time now. Technology is being developed every day to increase efficiency.It’s the endeavor of predicting what word comes straightaway.Long phrases might be tedious to write, but text prediction technology built into keyboards makes this simple. In addition, Next Word Prediction called ”Language Modeling.” The task at hand is predicting the first word that will be said. It has numerous applications and is one of the main tasks of human language technology. This method uses letter to letter prediction and says that it predicts a letter when letter is used to build a word. Long short time memory formula can sense past text and anticipate the words that can be useful for the user to border sentences. Word prediction software has been developed to help people talk more easily and to help those who write more slowly.

We train the algorithm for specific tasks and then use it in natural language processing, which will help solve some sentence generation problems, especially for application scenarios such as summary generation, machine translation, and automatic question answering. The BERT model is currently widely used language models for text generation and prediction.BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of NLP tasks.Word predictor has applications in various areas like texting, search engine etc. Our program uses a text file to predict the words which the user may think of thus helping a lot.

Synopsis

- **Title: Predicting Next Word in the sentence using BERT Algorithm**
- **Domain : Artificial Intelligence**
- **Sub-domain : Neural Network/Deep Learning**
- **Objectives:**

  1. By recommending the next word based on past text, this application seeks to minimise human effort.
  2. Using an BERT Algorithm to predict the appropriate word for the user's convenience.

- **Abstract:**

Writing long sentences is a little difficult, but text prediction technology built into the keyboard has made this simple. Language Modeling is another name for Next Word Prediction. It involves trying to predict the next word that will be spoken. It is one of the main tasks of human language technology and has a variety of uses. Long short time memory formula may recognise previous material and anticipate words that may be helpful for the user to border phrases. This method makes use of letter-to-letter prediction, which denotes that it anticipates a letter when a word is formed from other letters. Word prediction tools have been developed that could make it easier to talk and also help those who write more slowly.

- **Keywords:**

Natural Language Processing (NLP), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Language Modeling (LM)

- **Problem Definition:**

  Develop a graphical user interface based text prediction system, which suggests text based on the previous typed text by the user through recurrent neural network and Long Short Term Memory.

- **List of Modules:**

1. Client Interface
2. Prediction Program - RNN model
3. Model Dataset

- **Current Market Survey:**

1. VIPA : VIPA is a software platform that is designed to swallow, process and display large numbers of disparate streaming data flows, including video, audio, text, etc. It is developed by a company named Oceanit and it was founded in the year 1985. It has its primary application in various fields.
2. Linguamatics : Linguamatics is the world leader in deploying innovative natural language processing (NLP)-based text mining for high-value knowledge discovery and decision support. It is a private company and was founded in the year 2001. Linguamatics I2E is used by top commercial, academic and government organizations.

- **Scope of the Project:**

1. Every person using modern technology platforms needs to use the text prediction model based on BERT Algorithm.
2. This system has been significantly improved with useful features for future upgrades. BERT is the algorithm used to improve text that is predicted automatically.
3. Text prediction is widely used across a variety of software platforms, including webpages, computer programmes, and mobile apps. In order to anticipate the text, deep learning or machine intelligence employs a variety of supervised and unsupervised machine-learning methods.

- **Literature Survey:**

1. Next Word Prediction
   Author - Keerthana N, Harikrishnan S, Konsaha Buji M, Jona J B
   Year - 2021
   Summary - Suggests subsequent immediate word supported this out their word. These systems work victimization machine learning algorithms that has limitation to form correct syntax.

2. Recurrent Neural Network based Models for Word Prediction
   Author - S.Ramya, C.S.Kanimozhi Selvi
   Year - 2019
   Summary - Suggest and presented a comparative study on various models like Recurrent Neural Network, Stacked Recurrent Neural Network, Long Short Term Memory network (LSTM) and Bi-directional LSTM that gives solution for the above said problem.

3. Predicting next Word using RNN and LSTM cells: Stastical Language Modeling
   Author - Aejaz Farooq Ganai, Farida Khursheed
   Year - 2019
   Summary - The paper describes how some common structural next word predicting queries would be satisfactorily described inside model.

4. A Text Generation and Prediction System: Pre-training on New Corpora Using BERT and GPT-2
   Author - Yuanbin Qu, Peihan Liu, Wei Song, Lizhen Liu*, Miaomiao Cheng
   Year - 2020
   Summary - We train the machine for specific tasks and then use it in natural language processing, which will help solve some sentence generation problems.

5. Natural Language Word Prediction Model Based on Multi-Window Convolution and Residual Network

Author - Jingyun Yang , Hengjun Wang, Kexiang Guo

Year - 2020

Summary - Proposed MCNN-ReMGU model based on multi-window convolution and residual-connected minimal gated unit (MGU) network for the natural language word prediction.

- **Software and Hardware Requirement of the Project:** *Software:*

  1. VS code
  2. Keras
  3. Tensorflow
  4. NLTK(Natural Language Toolkit)

  Hardware:

  1. Processor : 2.6 GHz
  2. RAM : 4GB
  3. Hard Drive : 40GB

- **Contribution to Society:** Our Next Text Prediction Model provides a better platform for all text typers by automating text suggestion based on the user's identified pre-typed text. These assist users in fast typing and getting relevant text suggestions based on their pre-typed text, putting them into lesser efforts. A text prediction system improves people's user experience of typing and searching text across the internet.

- **Probable Date of Project Completion:** March 2023

- **Outcome of the Project:**

  1. The proposed model will suggest the text by predicting the next text with the help of : Deep Learning/Machine Learning (Recurrent Neural Network).
  2. Previously typed text will be analysed for further text prediction.

Abbreviation

RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
BERT	Bidirectional Encoder Representations from Transformers
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
ReLU	Rectified Linear Unit

List of Figures

List of Tables

INDEX

CHAPTER 1

INTRODUCTION

1.1 ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI)[7] is the emulation of human intelligence in devices that have been designed to behave and think like humans. Artificial intelligence is founded on the idea that human intelligence can be described in a way that makes it simple for a computer to duplicate it and carry out activities of any complexity. Artificial intelligence's objectives include imitating human cognitive function.

1.1.1 How does AI work?

As the hype around AI has accelerated, vendors have been scrambling to promote how their products and services use AI. Often what they refer to as AI is simply one component of AI, such as machine learning. AI requires a foundation of specialized hardware and software for writing and training machine learning algorithms[7]. Programming language is synonymous with AI, include Python, R and Java..

A lot of labelled training data is typically ingested by AI systems, which then examine the data for correlations and patterns before employing these patterns to forecast future states. Ability to reason and take actions that have the best likelihood of reaching a certain objective is the ideal quality of artificial intelligence. Machine learning (ML) is a subset of artificial intelligence. Deep learning algorithms enable this automatic learning by ingesting massive volumes of unstructured data, such as text, photos, or video.

1.2 DEEP LEARNING

Deep learning can be considered as a subset of machine learning. It is a field that is based on learning and improving on its own by examining computer algorithms[8] .While machine learning uses simpler concepts, deep learning works with artificial neural networks, which are designed to imitate how humans think and learn.

1.2.1 How Does Deep Learning Work?

Similar to how the human brain is made up of neurons, neural networks are layers of nodes. Individual layer nodes are linked to neighbouring layer nodes[8]. The number of layers in the network indicates how much deeper it is. Signals go between nodes and assign matching weights in an artificial neural network. A node with a higher weight will have a greater impact on the nodes in the layer below it. The weighted inputs are combined to create an output in the final layer. Because deep learning systems process a lot of data and include several intricate mathematical computations, they demand strong hardware.

Numerous fields, including pattern recognition, natural language processing, and computational learning, have seen extensive use of deep learning approaches.

1.3 NEURAL NETWORK

A neural network is a collection of algorithms that aims to identify underlying links in a set of data using a method that imitates how the human brain functions. Since neural networks are capable of adapting to changing input, they can produce the best possible results without having to change the output criterion[9].

Layers of connected nodes make up a neural network. Every node is a perceptron, which resembles a multiple linear regression. The multiple linear regression signal is fed into a potentially nonlinear activation function via the perceptron.

1.4 TRANSFORMER NEURAL NETWORK

A revolutionary design called the Transformer Neural Network aims to handle long-range dependencies with ease while resolving sequence-to-sequence problems. We shall comprehend why we use it and where it comes from before moving immediately to the topic of Transformer. Beginning the narrative is RNN (Recurrent Neural Networks). The Feed Forward Neural Networks, or RNNs, are implemented gradually.[10] RNNs are designed to take a series of inputs with no predetermined limit on size. Basic feedforward networks “remember” things too, but they remember the things they learnt during training. While RNNs learn similarly while training, in addition, they remember things learnt from prior input(s) while generating output(s) Unfortunately, as that gap grows, RNNs become unable to connect, as their memory fades with distance.

1.5 TRANSFORMERS

Transformer is an encoder-decoder architecture based on attention layers. One main difference is that the input sequence can be passed parallelly so that GPU can be utilized effectively, and the speed of training can also be increased. And it is based on the multi-headed attention layer, vanishing gradient issue is also overcome by a large margin[11] .

Now comes the main algorithm which we are going to use in our project BERT (Bidirectional Encoder Representations from Transformers)

1.5.1 How BERT works

BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore it is considered bidirectional, though it would be more accurate to say that it’s non-directional. This characteristic allows the model to learn the context of a word based on all of its surroundings (left and right of the word) [3].

The goal of any given NLP technique is to understand human language as it is spoken naturally. In BERT’s case, this typically means predicting a word in a blank. To do this, models typically need to train using a large repository of specialized, labeled training data. This necessitates laborious manual data labeling by teams of linguists[1].

BERT, however, was pre-trained using only an unlabeled, plain text corpus (namely the entirety of the English Wikipedia, and the Brown Corpus). It continues to learn unsupervised from the unlabeled text and improve even as its being used in practical applications (ie Google search). Its pre-training serves as a base layer of ”knowledge” to build from. From there, BERT can adapt to the ever-growing body of searchable content and queries and be fine-tuned to a user’s specifications. This process is known as transfer learning.

1.6 PROJECT IDEA

As we type in what is the weather we already receive some predictions. We can see that certain next words are predicted for the weather. When a user is texting or typing, the next word prediction can be fantastic. Understanding the user's messaging tendencies would save a tonne of time. Additionally, our virtual assistant might use this to finish some statements. Overall, we will be implementing the next word prediction feature of the predictive search system.

1.7 MOTIVATION OF THE PROJECT

When autocomplete successfully guesses the term a user intends to write after only a few characters have been entered into a text input box, it speeds up human-computer interactions. Its influence permeates every part of our life, from messaging to looking up information or files online. For some people, this is what enables them to perform even the most fundamental duties with ease, including self-expression, interpersonal communication, searching, and simple job-related tasks. The user would find the entire process to be much more convenient thanks to this project, and they may also save time and effort. By enhancing the effectiveness and speed of word prediction through our project, we sincerely hope to accelerate this procedure and provide the public with a solution to this specific problem.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Nusrat Jahan Prottasha et.al[1] compares and contrasts various machine learning and deep learning algorithms for classifying texts according to their topics. They have demonstrated how transfer learning, the new revolution in natural language processing, can surpass all previous architectures. They have shown that transformer models such as BERT with proper fine-tuning can play a crucial role in sentiment analysis. Additionally, a CNN architecture was developed for this classification task.

Min Kang et.al proposed method effectively performs data augmentation. The results show that the augmentation method based on filtered BERT improved the performance of the model. they suggests that their method can effectively improve the performance of the model in the limited data environment [2].

Yuanbin Qu et.al proposed the model by using different corpora, and used the trained model to complete the long sentence generation and masked word generation prediction task. The former mainly generates sentences by looping down from the start word, and the latter is based on the surroundings word to generate intermediate words. Through the experimental results, we can know that the GPT-2 and BERT models perform very well in text generation tasks [3].

Jingyun Yang et.al[4] proposed MCNN-ReMGU model based on multi-window convolution and residual-connected minimal gated unit (MGU) network for the natural language word prediction. First, the convolution kernels with different sizes are used to extract the local feature information of different graininess between the word sequences. Then, the extracted features are fed to the residual-connected MGU network. Finally, the prediction results are output by the SoftMax layer.

Aejaz Farooq Ganai et.al[5] introduced Language Modeling is defined as the operation of predicting next word. It is considered as one of the basic tasks of Natural Language Processing(NLP) and Language Modeling has several applications. In this research paper, the assorted potentialities for the efficient utilization of language models in structured document retrieval are mentioned. A tree-based generative language model for ranking documents and parts has been used here.Nodes within the tree correspond to different document parts like titles, paragraphs and sections.

S.Ramya et.al studied compared various sequential models with various loss functions and activation functions. Through this study the Bidirectional LSTM shows minimum loss compared to that of other models for the training dataset Combination of Bidirectional LSTM with Cross Entropy: Kullback-Liebler Divergencework better than Bidirectional LSTM with Cross Entropy: RelU.Among the three activation functions sigmoid, tanh and relu, the activation function relu work better[6].

Sr. No.	Paper Title	Year of Publication	Method Algorithm Used
1	Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning	2022	BERT
2	Similarity Filter-Based Augmentation with Bidirectional Transfer Learning	2021	Filtered BERT
3	A text Generation and Prediction System: Pre-training on New Corpora Using BERT and GPT-2	2020	BERT and GPT-2
4	Natural Language Word Prediction Model Based on Multi-Window Convolution and Residual Network	2020	LSTM
5	Predicting next Word using RNN and LSTM cells: Stastical Language Modeling	2019	RNN and LSTM
6	Recurrent Neural Network based Models for Word Prediction	2019	RNN

Table 2.1: Comparative Analysis

CHAPTER 3

PROBLEM DEFINITION AND SCOPE

3.1 PROBLEM STATEMENT

Develop a graphical user interface based text prediction system, which suggests text based on the previous typed text by the user through recurrent neural network and Long Short Term Memory.

3.1.1 Goals and objectives

Goal and Objectives:

- This application is focused on reducing human efforts by suggesting the next word based on previous text.
- Utilizing RNN neural network which will predict the relevant word for convenience of user.

3.1.2 Statement of scope

- In technology platforms, the text prediction model based on the Recurrent Neural Network system is extremely important for every person.
- This system has been significantly improved with useful features for future upgrades. Recurrent neural network is the technology used to improve text that is predicted automatically.
- Text prediction is widely used across a variety of software platforms, including websites, computer programs, and mobile apps. In order to predict the text, deep learning or machine intelligence employs a variety of supervised and unsupervised machine-learning algorithms.

3.2 SOFTWARE CONTEXT

The entirety of the numerous features a developer requires to finish a task can be referred to as a programming context. Programmers interpret the same information differently based on their context, which includes information from multiple sources goals for programming. A context model outlines the management and structure of context data. It aims to formalise or semi-formalize descriptions of the contextual data found in context-aware systems. In other words, the model offers the mathematical interface and behavioural description of the environment, while the context is the components of the system that surround it. utilised to display reusable contextual data for a component.

3.3 MAJOR CONSTRAINTS

- Information confidentiality
- A user friendly and interactive GUI which let the user, use the system effectively.
- For more precise and accurate results we need large dataset.

3.4 METHODOLOGIES OF PROBLEM SOLVING AND EFFICIENCY ISSUES

Using theory and research to identify solutions to issue domains, idea testing, and the use of best practises are all part of the problem solving process in software development. Utilizing logic and imagination to find problems and find software solutions is another aspect of problem-solving.

3.5 OUTCOME

- Provide option to select how many number of words to be predicted.
- Predict the next word in the sentences.
- Provide relevant output to the user.

3.6 APPLICATIONS

- Word processors, search engines, messaging services like WhatsApp, command-line translators, and more can all use Word Predictor.

- Word prediction software is developed to assist people with physical limitations boost their typing speed and reduce the number of keystrokes required to complete a word or a sentence.
- The next word prediction model is useful for the people who are victim of the disease known as Attention deficit hyperactivity disorder (ADHD) , it will help them to type efficiently.

3.7 HARDWARE RESOURCES REQUIRED

Sr. No.	Parameter	Minimum Requirement	Justification
1	CPU Speed	2 GHz	Minimum
2	RAM	4 GB	Minimum
3	Hard Disk	50 GB	Minimum

Table 3.1: Hardware Requirements

3.8 SOFTWARE RESOURCES REQUIRED

Platform :

1. Operating System: Windows OR Linux OR MAC
2. IDE: Microsoft Visual Studio
3. Programming Language : Python

CHAPTER 4

SOFTWARE REQUIREMENT SPECIFICATION

4.1 INTRODUCTION

A requirements analysis is the process used to determine the needs and expectations of a new product. This includes frequent communication with stakeholders and end users of the product to define expectations, resolve conflicts and document all key requirements .Requirements are gathered through surveys and through one-on-one communication with customers or end users to gain a thorough understanding of the requirements and the requirements that must be met for the system to function properly. In this requirements analysis process, all key stakeholders and end users of the system are first identified, then through individual interviews or focus groups, the complete functional and non-functional requirements are identified. There are many different kinds of requirements, so they should be grouped to avoid confusion. Requirements generally fall into three categories:

1. Functional Requirements-Functions that a product must fulfill.
2. Technical Requirements-Technical aspects that should be considered for successful implementation of the product.
3. Operational Prerequisites-Operations that must be performed on the backend for the product to function properly.

After the requirements have been classified into one of the specified groups, they are checked for feasibility and only these requirements are retained and the others are discarded.

4.1.1 Purpose and Scope of Document

A series of use cases that define how users will interact with the code are sometimes included in a software requirements specification, which is a detailed description of the behaviour of a system that needs to be constructed. Additionally, it has non-functional criteria as well. Nonfunctional requirements, such as performance engineering requirements, quality standards, or design limitations, place restrictions on the design or implementation. All criteria needed for the project’s development are listed in the software requirements specification document. We must have a complete and accurate grasp of the items that will be developed in order to determine the needs. This was created following thorough discussions with the project team and the client. A software requirements specification is a detailed explanation of the intended use and setting for software that is currently being developed.

There are many excellent definitions of System and Software Requirements Specifications that can be used as a solid foundation for defining a fantastic specification and pointing out flaws in earlier efforts. Additionally, there is a tonne of excellent information about developing strong specifications online. Lack of information about how to prepare specifications appropriately or even what should be included in specifications is not the issue.

4.1.2 Overview of responsibilities of Developer

- Supervising the project plan's testing, documentation, and training initiatives.
- Analysis of risk management is done.
- ensuring that the project is finished on schedule and within the given budget.
- Regularly monitor project progress and report to the guide.
- Work on creating the project timeline, budget, and plan.
- Give team members different responsibilities to do.
- Coordinate with team members.

4.2 FUNCTIONAL REQUIREMENTS

The primary functional need is our programme, for which we will use a programming language, that is Python. The dataset file must always be accessed by the system. The user will enter data as a word (partial or full). The words that match the letters you've typed so far will be shown on the screen. Every time a word prediction cycle is finished, the system prompts the user to input further words. If so, a new cycle begins; otherwise, the system halts.

4.3 NONFUNCTIONAL REQUIREMENTS

4.3.1 Performance Requirements

- The system should be interactive to users.
- The interface is simple and easy to use.
- System is user friendly, self-explanatory
- This system can be used by everyone.
- Speed: The system should be made as fast as possible to reduce response time.
- Throughput: The throughput should be as high as possible. We should be able to attain maximum output in minimum time.
- Resource Utilization: Resources are modified according to user requirements.

4.3.2 Safety Requirements

- Operation of regular updation for the dataset should take place.
- The system should be tough and not prone to breakdowns.

4.3.3 Security Requirements

- The administrators maintain the system as per the maintenance contract.
- The system has to be secure from attacks.
- In case of breakdown should be stabilized soon.

4.3.4 Software Quality Attributes

- Try to attain maximum reliability.
- Reliability will also be higher since we try to attain maximum accuracy.
- Maintain proper and updated dictionary files to improve reliability.
- The information provided in the dataset files should be correct.

- Minimize the errors.
- All operations will be done correctly to increase the level of accuracy.

4.4 SYSTEM REQUIREMENTS

4.4.1 Database Requirements

4.4.1.1 Hardware Requirements

- Minimum CPU – P3/AMD Athlon 1.0 GHz+
- Minimum Disk Space – 512MB
- Minimum Memory – 500MB
- Touch Screens/Keyboard

4.4.1.2 Software Requirements

- Natural Language Toolkit (NLTK)
- Python Flask Server
- Keras
- Tensorflow
- Numpy
- Streamlit
- Pytorch

4.5 ANALYSIS MODELS: SDLC MODEL TO BE APPLIED

Waterfall Model

The Waterfall Model is sequential design process, often used in Software develop- ment processes, where progress is seen as flowing steadily download through the phase of conception, Initiation, Analysis, Design, Construction, Testing, Produc-

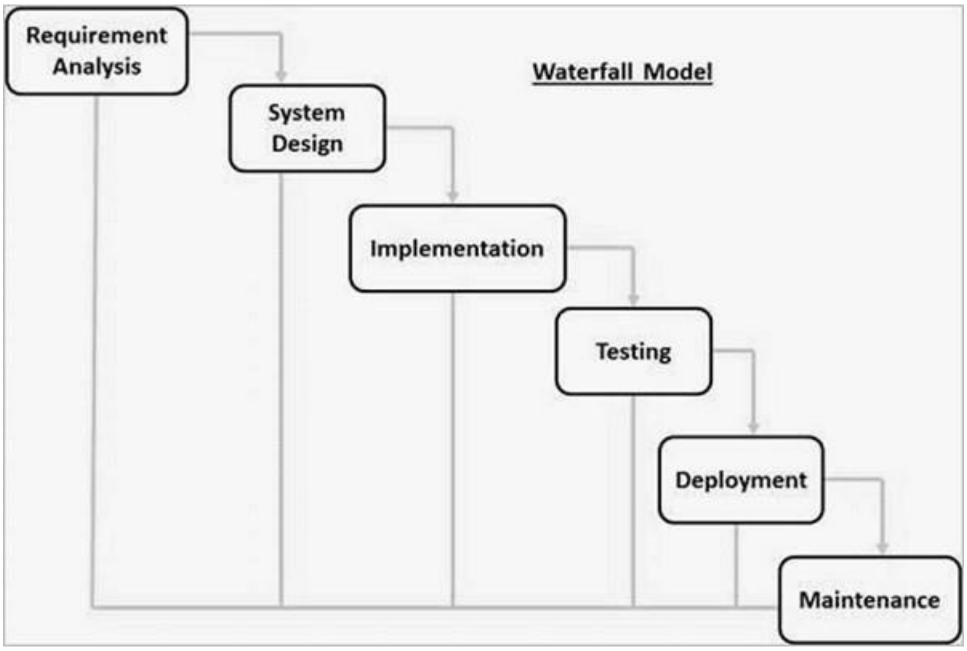

Figure 4.1: WaterFall Model

tion/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software de- velopments. This one of the oldest model followed in software engineering. There are 5 Phase of water fall model: Initiation, Analysis, Design, Construction, Testing, Produc- tion/Implementation and Maintenance. This Model is also called as the classic Life cycle model as it suggests a systematic sequential approach to software developments. The process begins with the communication phase where the customer specifies the re- quirements and then progress through other phases like planning, modeling, con- struction and deployment of the software . There are 5 Phase of water fall model:

1. COMMUNICATION

In communication phase the major task performed is requirement gathering which helps in finding out exact need of customer.y enables preventing a number of errors that could occur due to inadequate project control. It leads to widespread documentation development.

2. PLANNING

In planning major activities like planning for schedule, keeping tracks on the processes and the estimation related to the project are done. Planning is even used to find the types of risks involved throughout the projects.

3. MODELING

This is one the important phases as the architecture of the system is designed in this phase. Analysis is carried out and depending on the analysis a software model is designed.

4. CONSTRUCTION

The actual coding of the software is done in this phase. This coding is done on the basis of the model designed in the modeling phase.

5. DEPLOYMENT

In this last phase the product is actually rolled out or delivered installed at customer's end and support is given if required. A feedback is taken from the customer to ensure the quality of the product.

4.6 SYSTEM IMPLEMENTATION PLAN:

Month	Week	Date	Project Activity
August	1st Week	05/08/2022	Group Formation
August	2nd Week	12/08/2022	Project Undertaking
August	3rd Week	18/08/2022	Domain Selection
August	4th Week	26/08/2022	Study and Analysis of Domain
September	1st Week	10/09/2022	Finalization of Project Title
September	2nd Week	16/09/2022	Submission of Synopsis
September	3rd Week	23/09/2022	Feasibility Study
September	4th Week	30/09/2022	Review 1 and Review 2
October	1st Week	07/10/2022	Requirement Analysis
October	2nd Week	14/10/2022	System Architecture Design
October	3rd Week	21/10/2022	UML Diagrams Design
November	1st Week	04/11/2022	Changes in Uml Diagrams
November	1st Week	04/11/2022	Preparation for Review 3-4
November	2nd Week	11/11/2022	Review 3 and Review 4
November	3rd Week	18/11/2022	Prepared Project Stage 1 Report
November	4th Week	25/11/2022	Project Stage 1 Presentation

Table 4.1: Implementation Plan

4.7 SYSTEM IMPLEMENTATION GANTT CHART:

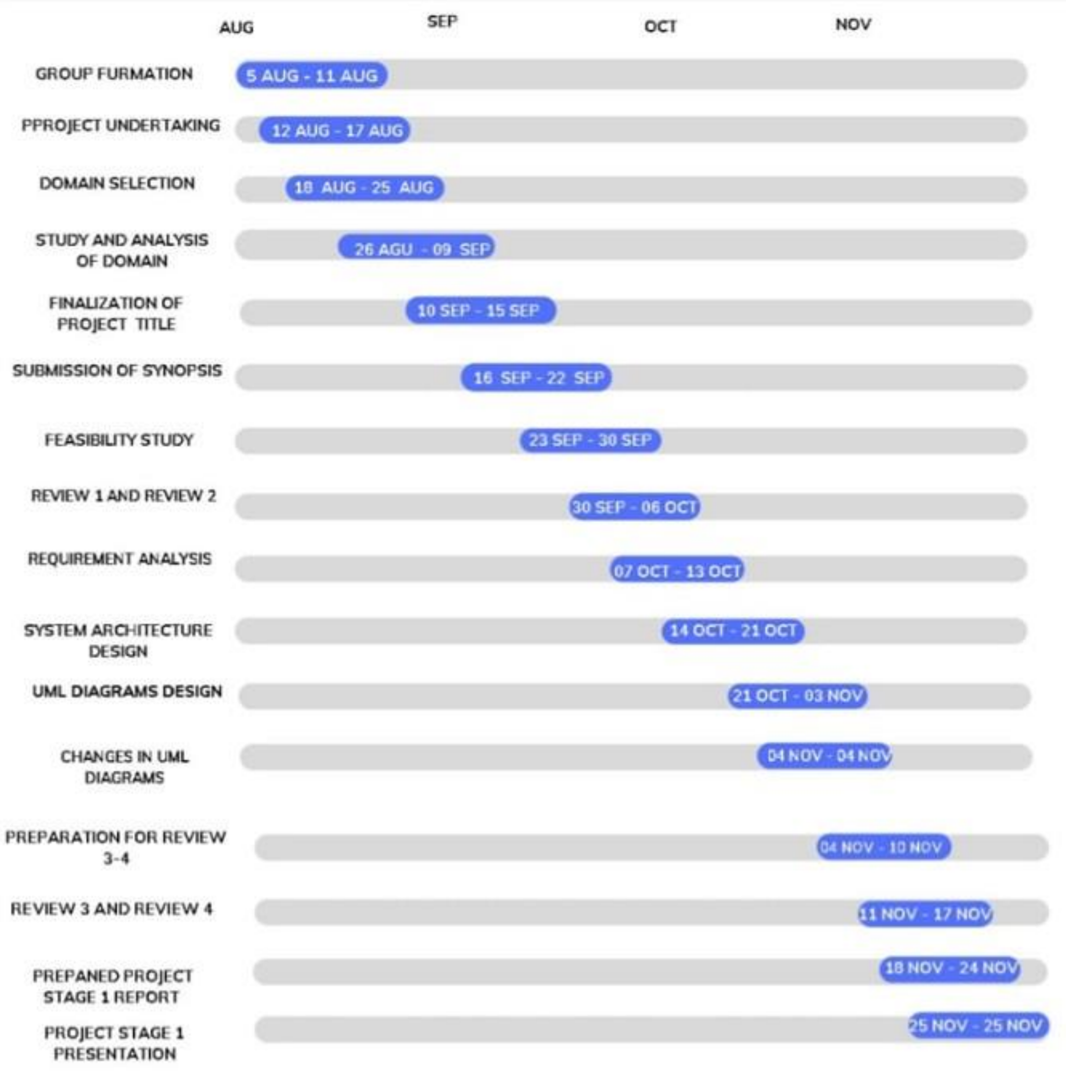

Figure 4.2: Gantt Chart

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

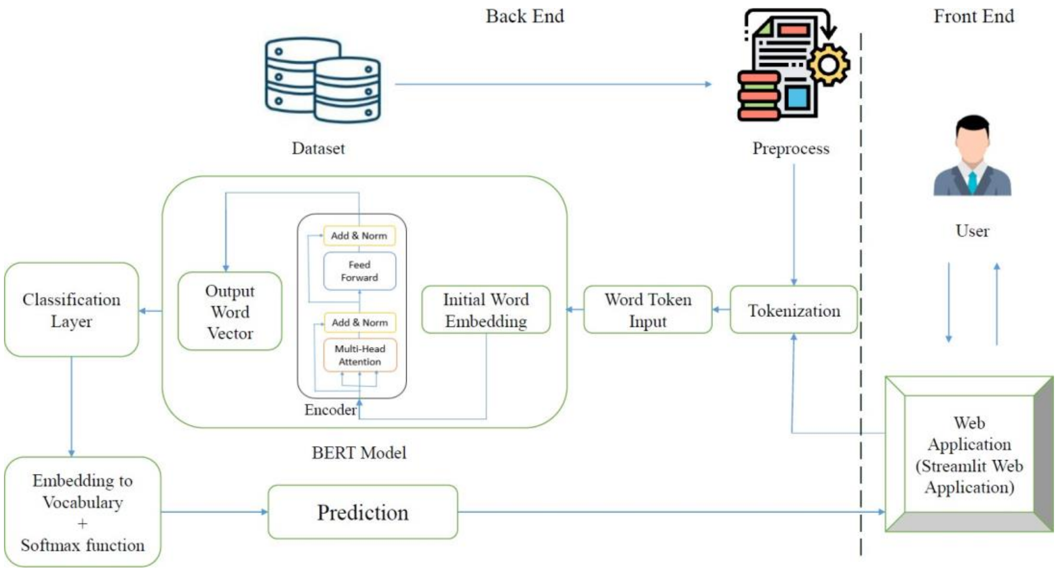

Figure 5.1: System Architecture

A software architecture is a set of principles that define the way software is designed and developed. An architecture defines the structure of the software system and how it is organized. It also describes the relationships between components, levels of abstraction, and other aspects of the software system. An architecture can be used to define the goals of a project, or it can be used to guide the design and development of a new system.

The above fig 5.1 show the major components of the system, which are

1.Data-Preprocessing :

A technique used to transform the raw data into useful data. Here we will remove punctuations marks, convert uppercase letters into lowercase letter

2.Tokenization:

Tokenization is breaking the raw text into small chunks. Tokenization breaks the raw text into words, sentences called tokens. These tokens help in understanding the context or developing the model for the NLP. The tokenization helps in interpreting the meaning of the text by analyzing the sequence of the words.

3.Encoder

In the encoding step, the Transformer uses learned word embedding to convert these words into word embedding vectors. Then they are passed into an attention-based encoder to generate the context-sensitive representation for each word. Eachword-embedding will have one output vector.

5.2 DATA FLOW DIAGRAM

DFD 0:

Level 0 Data flow diagram consist of basic overview of the flow diagram in which external entity as input that is sentences and output as the relevant words predicted.

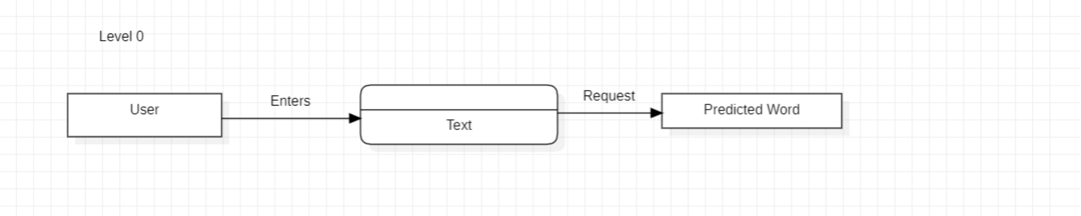

Figure 5.2: Data Flow Diagram Level 0

DFD 1:

First level data flow diagram consist of small deep overview of the system working flow that is in process we include Bidirectional Encoder Representation from Transformer. Here user can enter the sentences or the paragraphs to predict the further relevant text. Here the previous data will be taken as input and processed to generate the required output in the model.

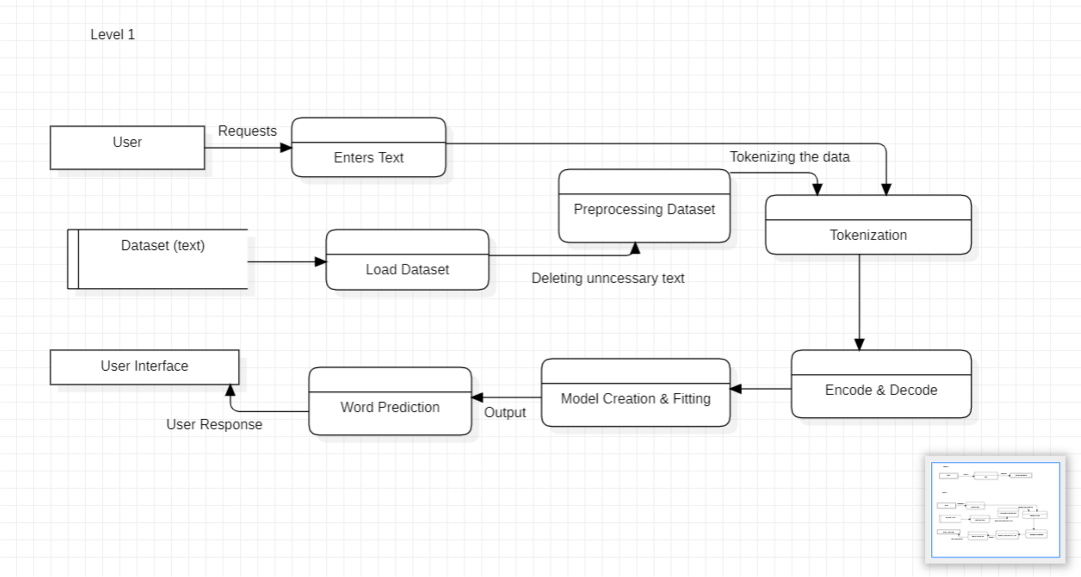

Figure 5.3: Data Flow Diagram Level 1

5.3 ENTITY RELATIONSHIP DIAGRAM

ER Diagram: Entity-Relationship model is referred to as an ER model. This data model is on a high level. The data items and relationships for a given system are defined using this model. An Entity Relationship Diagram is a diagram that shows the relationships between various entities in a system. It creates the database's conceptual design. Additionally, it creates a very straightforward and straightforward data view. The database structure is represented by an entity-relationship diagram in ER modeling.

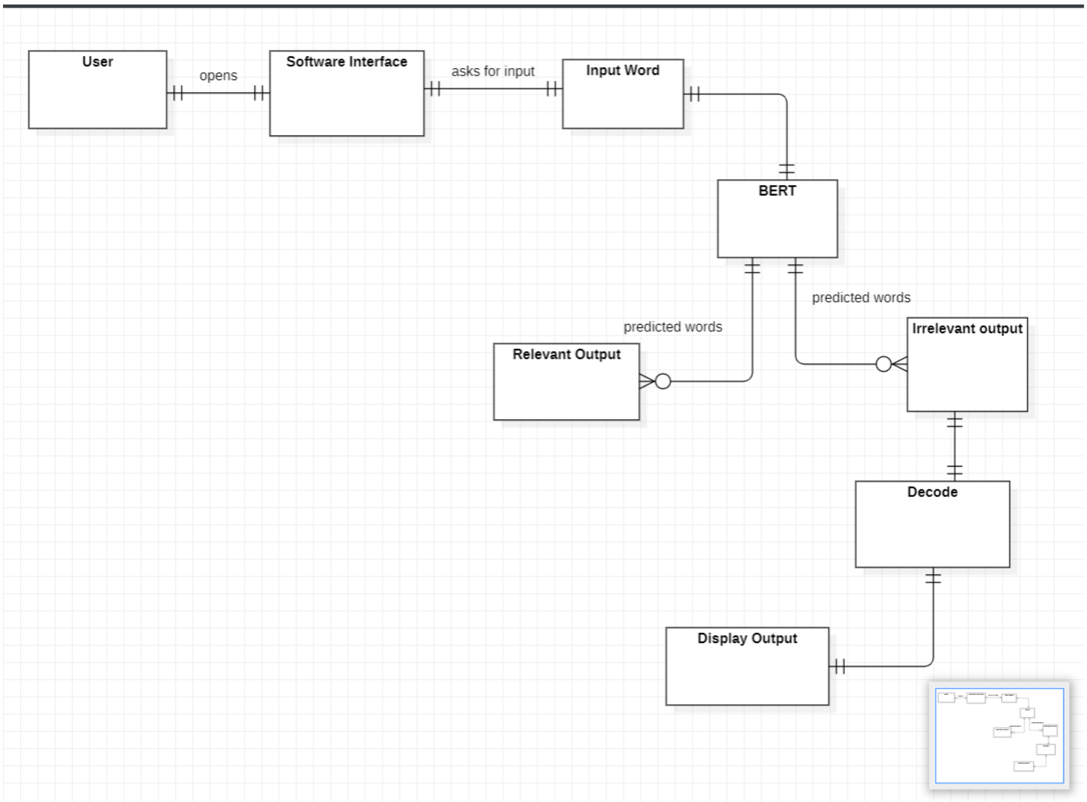

Figure 5.4: Entity Relationship

The user can enter a words and sentences in the input section and the user will have different option to select number of words to be predicted via model . The BERT model will tokenize the sentences into small chunks of words vector and provide them vector id . The probability distribution stage will find out relevant words to be predicted according to requirement . The irrelevant words will be decoded by decoder . Now , the relevant output will be displayed on the user interface .

5.4 UML DIAGRAMS

5.4.1 Class Diagram

Class Diagram:
Class diagrams describe the static structure of a system, or how it is structured rather than how it behaves. These diagrams contain the following elements: 1. Classes: which represent entities with common characteristics or features. These features include attributes, operations, and associations. 2. Relationships: which represent relationships that relate two or more other classes where the relationships have common characteristics or features. These features include attributes and operations.

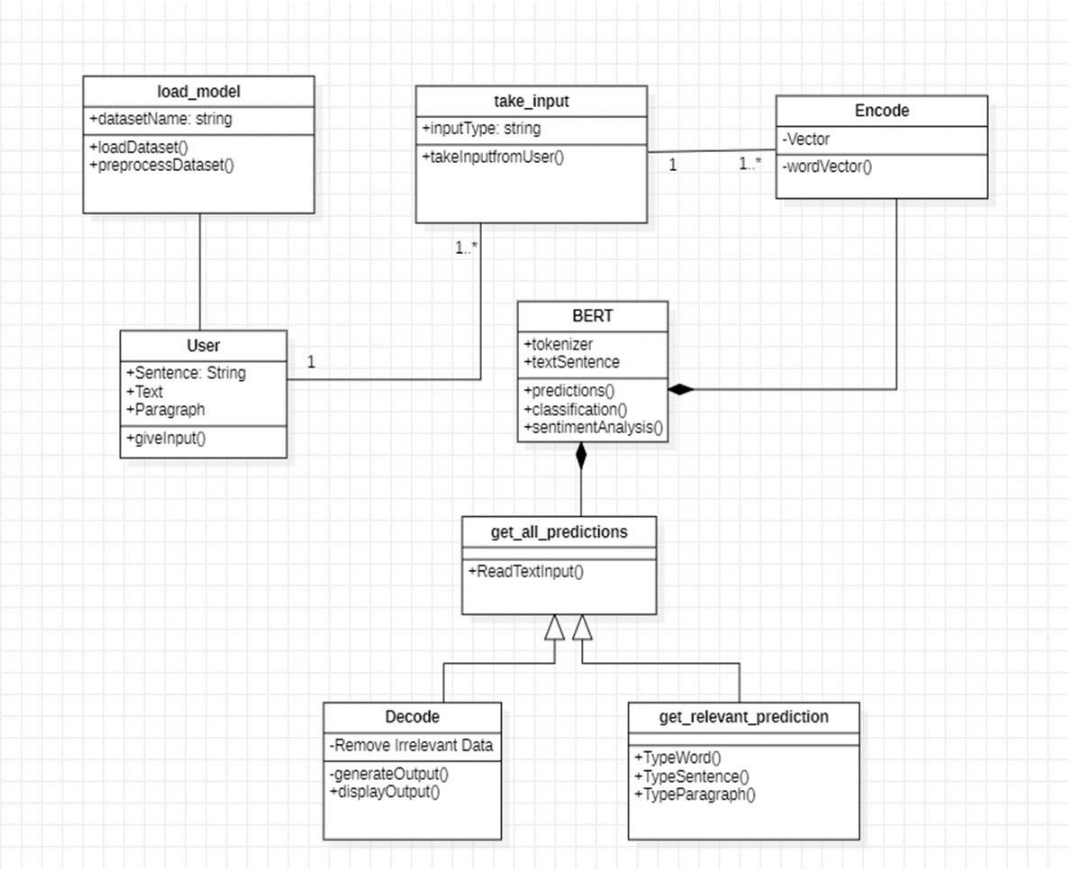

Figure 5.5: Class Diagram

In the model Eight different classes are created which are load- model , take-input , User , Encoder , BERT , Decode , get-all-prediction and get-relevant-prediction. They have some attributes and operations. The user has the opportunity to specify the number of words to be predicted by the model by entering words and phrases in the input section. The sentences will be tokenized using the BERT model into manageable word chunks and given a vector id and predicted word.

5.4.2 State Diagram

State transition diagrams provide a way to model the various states in which an object can exist. While the class diagram shows a static picture of the classes and their relationships, state transition diagrams model the dynamic behaviour of a system in response to external events (stimuli).A state diagram is a type of used in computer science and related fields to describe the behavior of systems.State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

In the state diagram of this model different states are presented according to the working of the system . The states involved in the model are input state, reading text state, Encoder state,BERT state etc. The model also have a state which will allow user to select how many number of words should be predicted according to requirements.

Every user has the choice to specify the amount of words to be predicted by the model by entering words and phrases in the input section. The sentences will be tokenized using the BERT model into manageable word chunks and given a vector id. The probability distribution step will identify pertinent words that can be predicted based on the specifications. The decoder will decode the meaningless words. The appropriate results will now be shown on the user interface.

State diagrams are employed to provide an abstract explanation of a system's behaviour. A series of events that can take place in one or more potential states are used to assess and illustrate this behaviour. By doing so, each diagram typically represents items of a single class and tracks the multiple states of its objects through the system.

It often proves to be a useful technique for modelling how the system and external entities interact and collaborate. To manage an object's state, it models event-based systems. Additionally, it specifies a number of unique states in which a system component may exist. Every item or component is in a particular state.

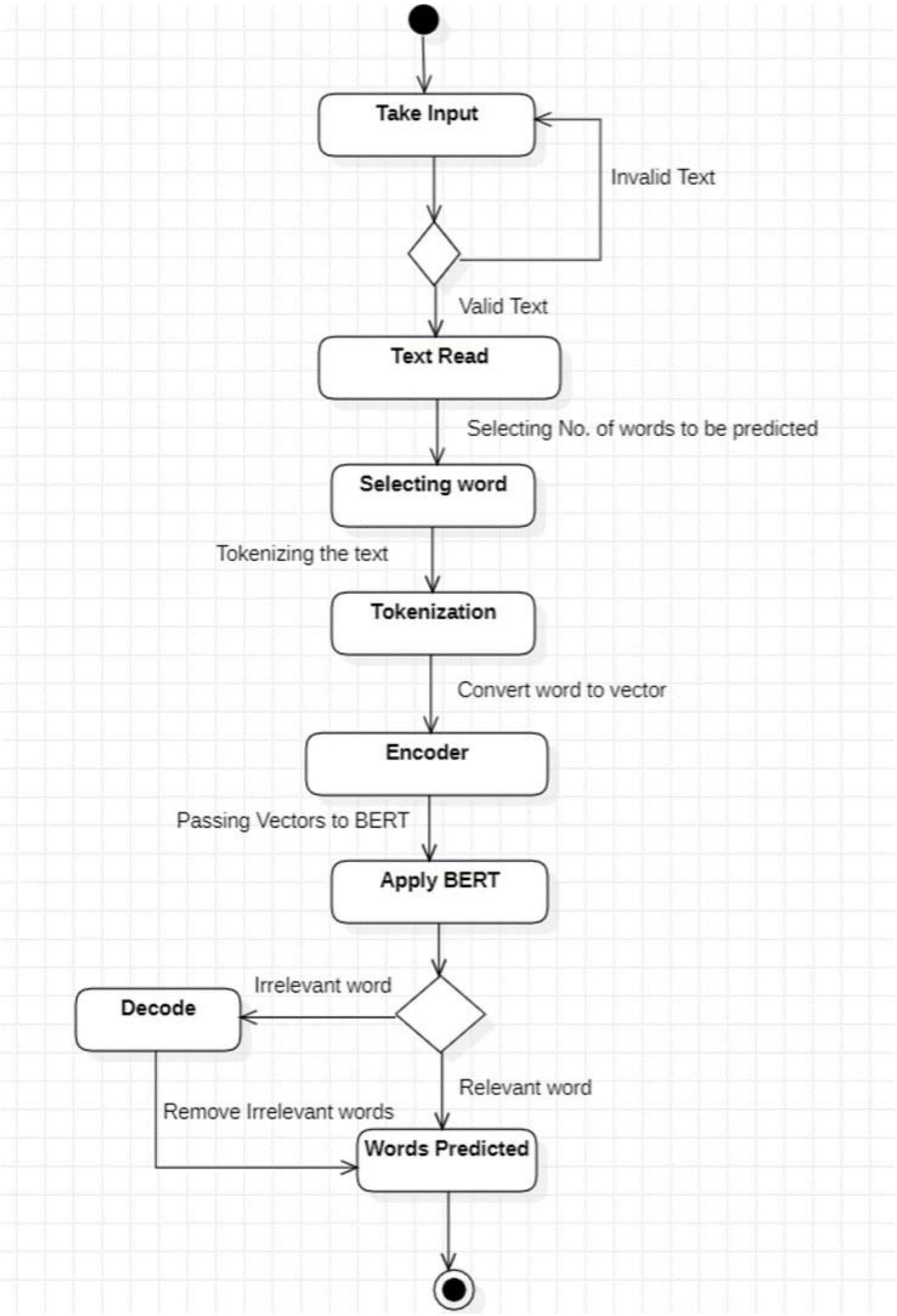

Figure 5.6: State Diagram

5.4.3 Use Case Diagram

Use case diagram: Use case diagrams describe the functionality of a system and users of the system. They contain the following elements:

1. Actors: which represent users of a system, including human users and other systems.
2. Use cases: which represent functionality or services provided by a system to users.

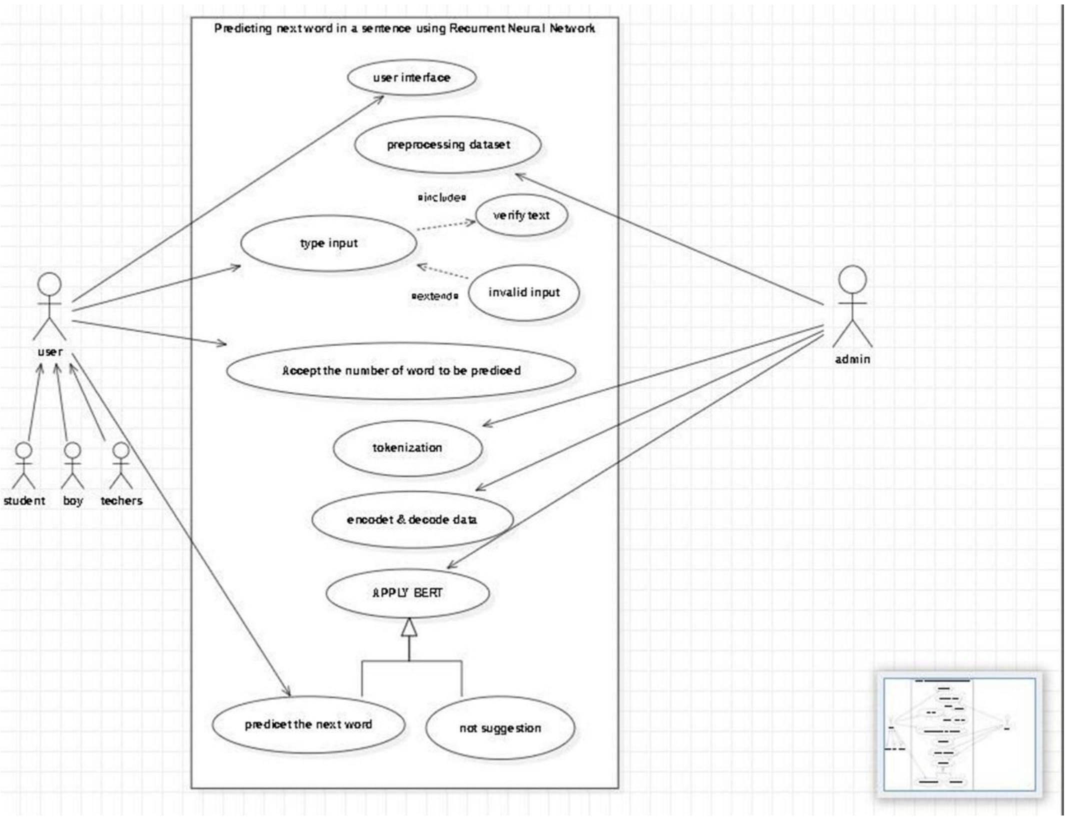

Figure 5.7: Use Case Diagram

The use case diagram of this model includes different use cases that are involved at the functioning of the system. The actors involved in the system are users which can be anyone and the second person is the admin who will manage the dataset and different activities. The different use cases involved in the system are text input, read text, perform diferent operations in the BERT model to get the output and at final stage display the predicted words as a result. The use cases are related with the actors according to the actions they perform.A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.

5.4.4 Activity Diagram

Activity Diagram:
Activity diagrams describe the activities of a class. They are similar to state transition diagrams and use similar conventions, but activity diagrams describe the behaviour/states of a class in response to internal processing rather than external events. They contain the following elements:

1. Action States: which represent uninterruptible actions of entities, or steps in the execution of an algorithm.
2. Action Flows: which represent relationships between the different action states on an entity.

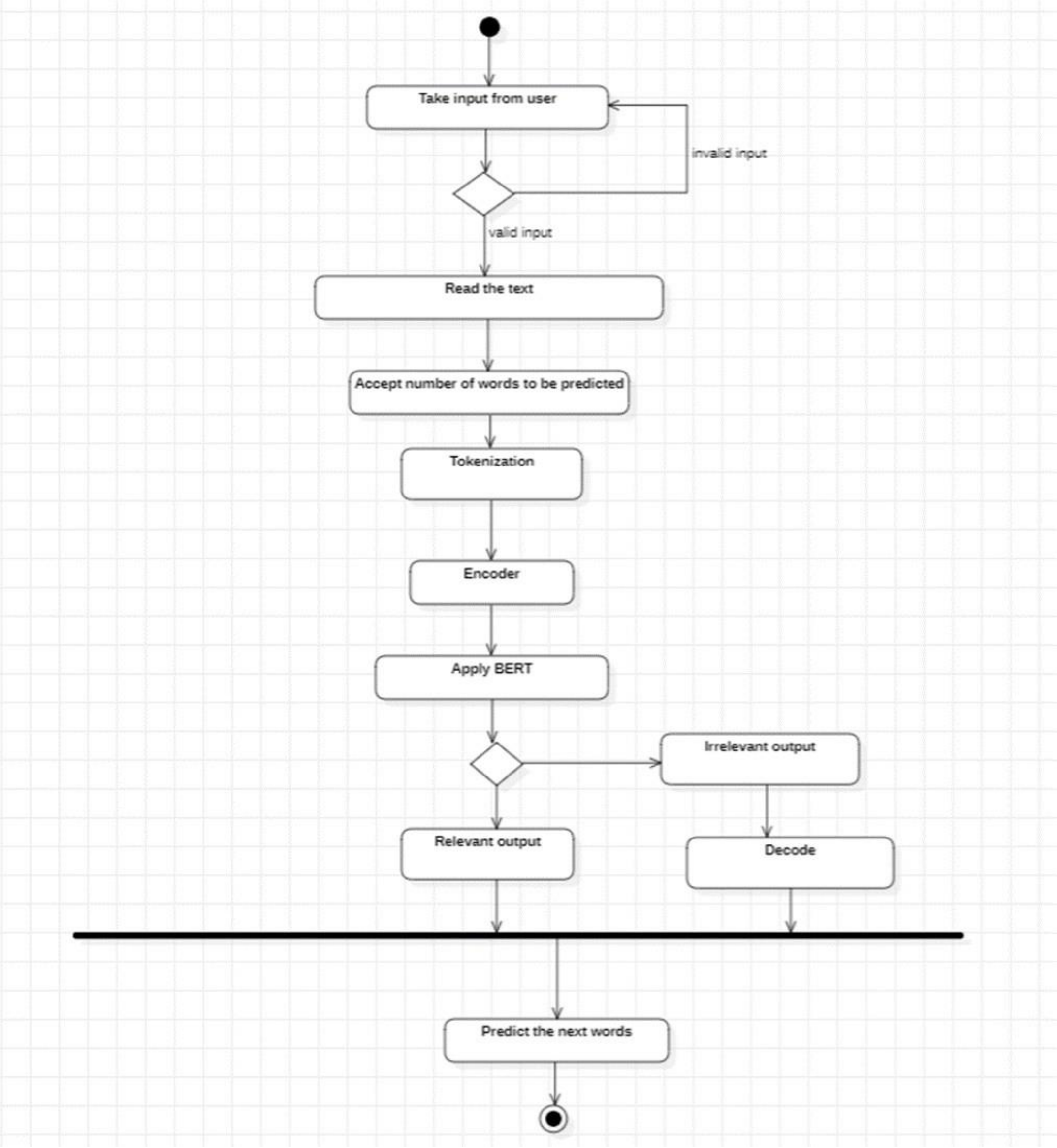

Figure 5.8: Activity Diagram

In the activity diagram of this model different activities are shown that are included in the system which are carried out to get relevant output as per requirement.

5.4.5 Sequence Diagram

Sequence diagrams typically show the flow of functionality through a use case. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. This diagrams show the events that external actors generate, their order, and possible inter-system events.All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main success scenario of the use case, and frequent or complex alternative scenarios. It consist of the following components:

1. Actors: involved in the functionality.
2. Objects: that a system needs to provide the functionality
3. Messages: which represent communication between objects.

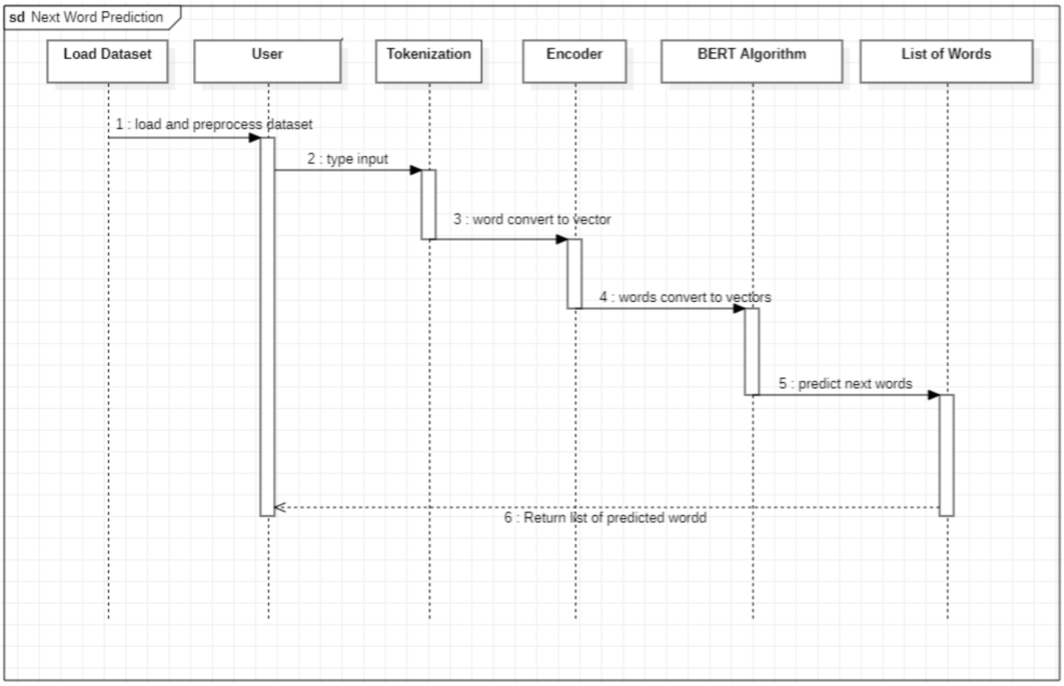

Figure 5.9: Sequence Diagram

The sequence diagram of this model states about the sequence of the task that are going to take during the model implementation. The task arranged according to the sequence are loading of the dataset, involvement of the user to give input, the process of tokenization which converts paragraphs and sentences into small chunks, the encoder to convert the words to vectors, BERT algorithm will predict the most relevant words and finally displaying the words that are predicted on the screen. It consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction.

CHAPTER 6

OTHER SPECIFICATION

6.1 ADVANTAGES

- Word prediction is a piece of assistive technology that offers word suggestions.
- This technology can help kids who have writing difficulties.
- Significantly improved performance compared to legacy methods.
- A simple method for utilising pre-trained models (transfer learning)
- The ability to tailor your data to the particular language environment and issue you are facing.

6.2 LIMITATIONS

- It is expensive. It requires more computation because of its size, which comes at a cost.
- It is designed to be input into other systems (not a standalone program), and because of that, it has to be fine-tuned for downstream tasks, which can be fussy.

6.3 APPLICATIONS

- Specifically for ADHD (Attention deficit hyperactivity disorder) victims
- Word processors, search engines, messaging services like WhatsApp, command-line translators, and more can all use Word Predictor.
- Word prediction software is developed to assist people with physical limitations boost their typing speed and reduce the number of keystrokes required to complete a word or a sentence.

Summary and Conclusion

Summary :

Communication is one of the most important characteristics of humanity. We communicate with one another mostly through language. How can our brain digest language so efficiently? How do words work in communication, and how are they read and understood? These are the only immediate, essential concerns that might concern us. I won't go into great detail on the neuroscience and neurolinguistics aspects, but I would want to encourage all readers who are interested to read the reference materials given in the reference section. Remaining on topic, NLP focuses on how computers can process, analyse, and interpret vast volumes of natural human language data, or more specifically, how computers interact with human language.

Modern, cutting-edge networks enable machines to replicate and learn from human-like tasks. In order to demonstrate how these language models may predict the following collection of words for the given input text, we shall cover various key network architectures.

An important area of natural language processing is the creation of sentences from provided starting words or the completion of incomplete phrases. It shows, in one way, whether a machine is capable of human thought and creativity. In order to tackle some problems with sentence production, we train the machine for specific tasks before using it in natural language processing. This is especially useful for application situations like summary generation, machine translation, and automatic question answering.

Currently, the BERT models are a popular language model. for the prediction and creation of text. The impressive performance of this model in the area of text production has been supported by numerous experiments. To improve its capacity for linguistic comprehension, BERT is trained on a variety of different tasks. Future Sentence Prediction With BERT BERT has developed three methods to anticipate the next sentence: In the first, sentences are used as input, while the output is a single class label, like in the case of the job below: The MNLI is an important classification task (Multi-Genre Natural Language Inference). The goal is to assess if the second assertion supports, refutes, or is neutral with respect to the first. Natural Language Inference Question (QNLI): The following actions by the model are required for this task:

Identify whether the second statement provides an answer to the query posed in the first. It is necessary to assess whether the second statement. whether the second builds on the first or not. The second type just requires one sentence as input, but the result is the same as the label for the next class. The task and data sets utilised for it are as follows: The Stanford Sentiment Treebank, or SST-2: It is a binary sentence classification job that uses sentences that were taken from movie reviews and annotated to show how they felt. This algorithm’s objective is to anticipate words in a phrase based on context.

Conclusion :

Word processors, web search engines, messaging services like WhatsApp, command-line translators, and more can all use Word Predictor. Word prediction software was initially developed to assist people with physical limitations boost their typing speed and to lessen the number of keystrokes required to complete a word or a sentence. Thus, utilising the Bert algorithm, we created our own word prediction tool that unquestionably improves user productivity.

References

[1] Prottasha, Nusrat Jahan, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud, and Mohammed Baz. ”Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning.” Sensors 22, no. 11 (2022): 4157.

[2] Kang, Min, Kye Hwa Lee, and Youngho Lee. ”Filtered BERT: similarity filter-based augmentation with bidirectional transfer learning for protected health information prediction in clinical documents.” Applied Sciences 11, no. 8 (2021): 3668.

[3] Yuanbin Qu, Peihan Liu, Wei Song, Lizhen Liu, and Miaomiao Cheng. A text generation and prediction system: Pre-training on new corpora using bert and gpt-2. In 2020 IEEE 10th international conference on electronics information and emergency communication (ICEIEC), pages 323–326. IEEE, 2020.

[4] Jingyun Yang, Hengjun Wang, and Kexiang Guo. Natural language word prediction model based on multi-window convolution and residual network. IEEE Access, 8:188036–188043, 2020.

[5] Aejaz Farooq Ganai and Farida Khursheed. Predicting next word using rnn and lstm cells: Stastical language modeling. In 2019 Fifth International Conference on Image Information Processing (ICIIP), pages 469–474. IEEE, 2019.

[6] S Ramya and CS Kanimozhi Selvi. Recurrent neural network based models for word prediction.

[7] McCarthy, John. ”What is artificial intelligence.” URL: http://www-formal. stanford. edu/jmc/whatisai. html (2014).

[8] Wolansky, Ivan. ”A deep dive into the basics of deep learning.” Proceeding of the Shevchenko Scientific Society. Medical Sciences 65.2 (2021).

[9] Pauliutkin, P. S. ”How do artificial neural networks work?.” (2022).

[10] Rani Horev. Bert explained: State of the art language model for nlp. Towards Data Science, 10, 2018.

[11] Zhang, Yuwen, X. Ding, Y. Liu, and P. J. Griffin. ”An artificial neural network approach to transformer fault diagnosis.” IEEE transactions on power delivery 11, no. 4 (1996): 1836-1841.

CHAPTER 7

PROBLEM STATEMENT FEASIBILITY

OBJECTIVE :

- It will identify the class of our problem.
- To find feasible solution to make problem statement from NP-hard to NP-Complete

Economic Feasibility :

Next word prediction is based on Transformer neural network architecture. It would not demand cost for software as most of its resources are open source and free to use. For small scale implementation dataset is freely available.

Technical Feasibility :

Software and Hardware used are easily available at no cost. Our model is trained on paragraph but in real life we can implement it by training it on large datasets. For large scale implementation we will use user's search history.

Social Feasibility :

The project is made by considering all types of people. Its is going to specially help victims of ADHD. Government can also embed this project into their systems. It will be beneficial in many areas. It is beneficial to society and has a broad scope.

By taking into consideration , we found out that our project is completely feasible, practical and reliable in all terms(technically ,economically, legally).

CHAPTER 8

DETAILS OF THE PAPERS REFERRED

1. Prottasha, Nusrat Jahan, Abdullah As Sami, Md Kowsher, Saydul Akbar Murad, Anupam Kumar Bairagi, Mehedi Masud, and Mohammed Baz. ”Transfer Learning for Sentiment Analysis Using BERT Based Supervised Fine-Tuning.” Sensors 22, no. 11 (2022): 4157.

2. Kang, Min, Kye Hwa Lee, and Youngho Lee. ”Filtered BERT: similarity filter-based augmentation with bidirectional transfer learning for protected health information prediction in clinical documents.” Applied Sciences 11, no. 8 (2021): 3668.

3. Yuanbin Qu, Peihan Liu, Wei Song, Lizhen Liu, and Miaomiao Cheng. A text generation and prediction system: Pre-training on new corpora using bert and gpt-2. In 2020 IEEE 10th international conference on electronics information and emergency communication (ICEIEC), pages 323–326. IEEE, 2020.

4. Jingyun Yang, Hengjun Wang, and Kexiang Guo. Natural language word prediction model based on multi-window convolution and residual network. IEEE Access, 8:188036–188043, 2020.

5. Aejaz Farooq Ganai and Farida Khursheed. Predicting next word using rnn and lstm cells: Stastical language modeling. In 2019 Fifth International Conference on Image Information Processing (ICIIP), pages 469–474. IEEE, 2019.

6. S Ramya and CS Kanimozhi Selvi. Recurrent neural network based models for word prediction.

7. Zhang, Yuwen, X. Ding, Y. Liu, and P. J. Griffin. ”An artificial neural network approach to transformer fault diagnosis.” IEEE transactions on power delivery 11, no. 4 (1996): 1836-1841.

8. Yu, Yong, et al. ”A review of recurrent neural networks: LSTM cells and network architectures.” Neural computation 31.7 (2019): 1235-1270.