**PROFORMA FOR THE APPROVAL PROJECT PROPOSAL**

###### (Note: All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)

PNR **No.: ……………………** Roll no**:**

* 1. Name of the Student
  2. Title of the Project
  3. Name of the Guide
  4. Teaching experience of the Guide
  5. Is this your first submission? Yes No

Signature of the Student Signature of the Guide

Date: ………………… Date: …………………….

Signature of the Coordinator Date: ………………

**PROJECT TITLE**

**A Project Report**

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

**Your Full Name (refer marksheet)**

**Your Roll No (A704)**

**Under the esteemed guidance of**

**Prof. & Prof.**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CHIKITSAK SAMUHA’S**

**S.S & L.S PATKAR COLLEGE OF ARTS & SCIENCE & V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS.**

**An Autonomous College**

**Affiliated To University Of Mumbai**

**Goregaon (W), Mumbai – 400 062**

**CHIKITSAK SAMUHA’S**

**S.S & L.S PATKAR COLLEGE OF ARTS & SCIENCE & V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS.**

**An Autonomous College**

****

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the project entitled, **"Project Name"**, is bonafide work of **Your Name** bearing Seat.No: submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

**Internal Guide Coordinator**

**External Examiner**

**Date: CollegeSeal**

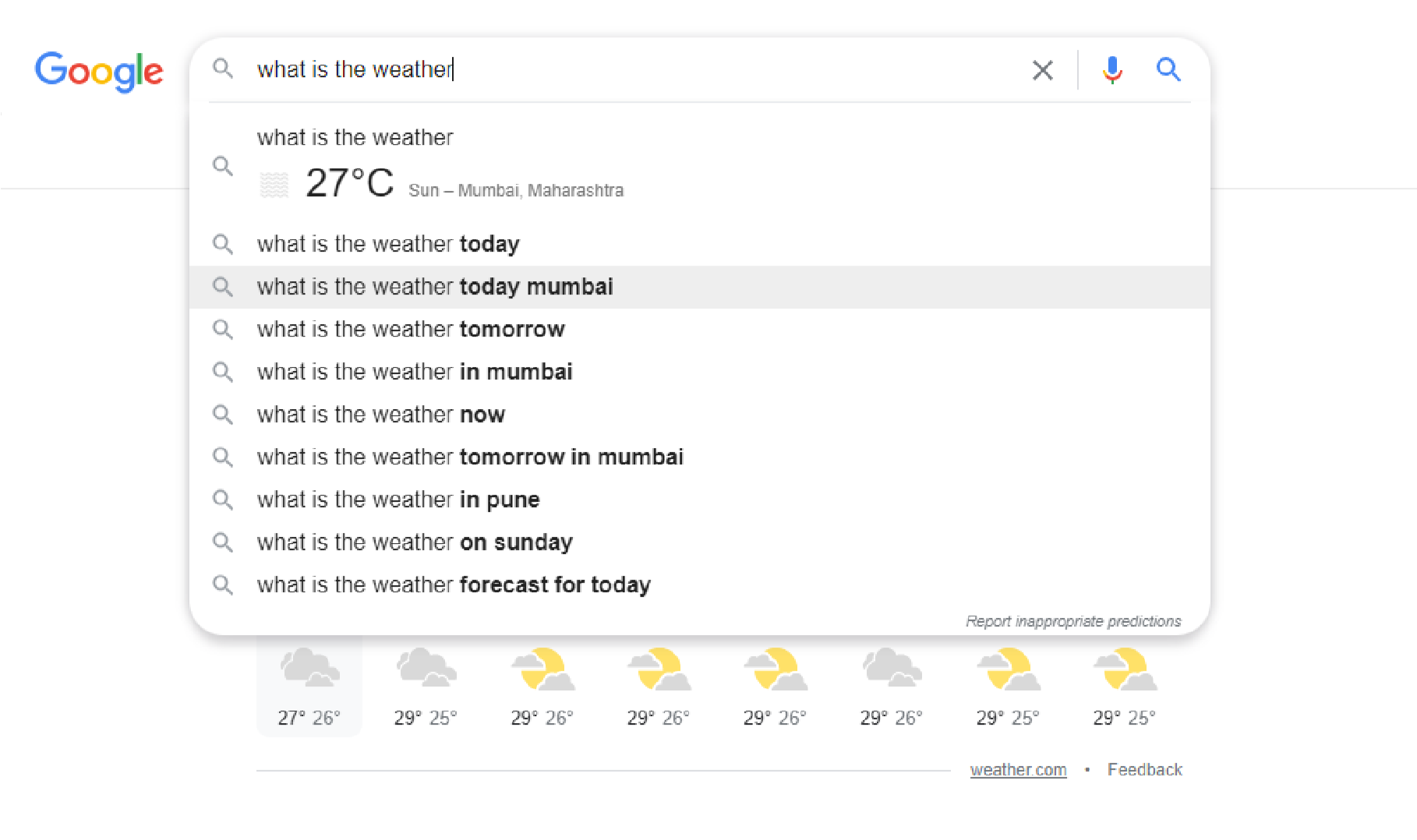
**COMPANY CERTIFICATE**

(ONLY FOR LIVE PROJECTS)

IF APPLICABLE

**ABSTRACT**

#### Wouldn’t it be cool for your device to predict what could be the next word that you are planning to type? This is similar to how a predictive text keyboard works on apps like WhatsApp, Facebook Messenger, Instagram, e-mails, or even Google searches. Below is an image to comprehend these predictive searches.



The next word prediction for a particular user’s texting or typing can be awesome. It would save a lot of time by understanding the user’s patterns of texting.

A next word prediction project is aimed at developing a machine learning model that can predict the next word in a sequence of words based on the previous words. The project uses a combination of deep learning models, such as LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) networks, to analyze the patterns and relationships in the input text data. The goal of the project is to develop a highly accurate and efficient model that can be used in a variety of applications, such as natural language processing, text completion, and language translation. The project involves collecting and pre-processing a large amount of text data, training the LSTM and GRU models on this data, and evaluating the performance of the models using various evaluation metrics. The final outcome of the project is a fully integrated and tested system that can be deployed and used to predict the next word in a sequence of words with high accuracy.

**ACKNOWLEDGEMENT**

### DECLARATION

I hereby declare that the project entitled, “**Project Name**” done at **College Name**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as a final semester project as part of our curriculum.

**Name and Signature of the Student**

**Date :**

**TABLE OF CONTENTS**

* **Chapter 1 Introduction**.......................................................................................... **00**
* Theoretical Background
* Objectives of the Project
* Purpose, Scope and Applicability of the Project
* Expected Achievements
* Organization of Report

* **Chapter 2 Survey of Technologies**.……………………………………………..…**00**
* Description of Available Technologies
* Comparative Analysis of Technologies in Chosen Area
* Chosen Project Domain
* Technologies to be used
* Reason Supporting the use of above selected technologies
* **Chapter 3 Requirements & Analysis**……………………………………………...**00**
* Problem Statement & Definition
* Requirements Specification
* Feasibility
* Planning and Scheduling
* Preliminary Product Description
* Conceptual Model
* **Chapter 4 System Design**………………………………………………………… .**00**
* Basic Modules
* Data Design
* Procedural Design
* User Interface Design
* Security Issues
* Dataset
* **Chapter 5 Implementation and Testing**…………………………………………… .**00**
* Implementation Approaches
* Coding Details and Code Efficiency
* Testing Approach
* Test Cases
* Modification and Expected Improvements
* **Chapter 6 Results and Discussions**……….…………………………..…………… .**00**
* Test Reports
* User Documentation
* Cost Estimation
* **Chapter 7 Conclusions**………………………………………………..……..……… .**00**
* Conclusion
* Limitations
* Future Scope of the Project
* **References**………...…………………………………………………..……….…… .**00**

**LIST OF TABLES**

Table 1.1 Name of the table…………………………………………………..00

**LIST OF FIGURES**

Fig 1.1 Name of the figure………………………………………………………….00

(Source Citation of the figure must be done on the references page)

**CHAPTER 1 : INTRODUCTION**

1. **Theoretical Background**

#### Before this technology came into the world, writing long emails used to be time consuming and there also used to be many spelling and grammar errors. Sometimes you don't know exactly what word would perfectly fit in a particular sentence. People who have dyslexia or any other difficulties in typing or word recall now have an efficient way to communicate.

#### Kids who have trouble spelling can broaden the range of words they use while writing. Without word prediction, these kids often avoid words they can't spell. Kids with writing issues can use word prediction as extra support, so they can focus on the ideas they're trying to express. Word prediction can be a big help for many people with learning and thinking differences.

#### Next word prediction comes in to solve all these problems. You can now work much faster and efficiently when writing emails or sending texts. It understands your patterns and gives you the next possible word after the last word of each sentence. Word prediction software may improve writing fluency for students whose typing is slow and laborious.

1. **Objectives of the Project**

Start to write from here

* Develop a machine learning model that can accurately predict the next word in a sequence of words based on the previous words.
* Evaluate the performance of the model by comparing its predictions against the actual next words in the sequence.
* Investigate the suitability of different deep learning models, such as LSTM and GRU, for next word prediction tasks.
* Pre-process the text data to ensure that it is suitable for analysis and to improve the performance of the models.
* Implement the models in a user-friendly application that allows users to input text and receive predictions of the next word.

1. **Purpose, Scope & Applicability of the Project**
   1. **Purpose**

Start to write from here

* Text completion: A next word prediction model can be used to complete a partially written sentence or text, making it easier and faster for users to express their ideas.
* Natural language processing (NLP): A next word prediction model can be used as a component in larger NLP systems, such as machine translation or question answering systems, to generate or select the next word.
* User input optimization: A next word prediction model can be integrated into user interfaces, such as text editors or messaging apps, to improve the user experience and reduce typing effort.

Overall, the purpose of a next word prediction project is to leverage the capabilities of machine learning to create more efficient and intelligent text-based applications that enhance human communication and productivity.

* 1. **Scope**

Start to write from here

* Limitation
* Restricted to dataset
* Data Quality
* Hyper parameter Tuning
* Computational Resources
* Accessibility

Accessibility refers to the degree to which a product, service, or technology can be used by people with disabilities or other limitations. In the context of a next word prediction project, accessibility can be considered in terms of the usability and user-friendliness of the model's interface, as well as its ability to support individuals with disabilities.

This project would be very helpful to everyone who write emails and texts regularly which is something done by everyone today

* 1. **Applicability**

Start to write from here

* In today’s world, next word prediction is an assistance system for typing.

● It reduces efforts and lessens time-consuming during typing.

* Besides that, some people have difficulty with slow typing speed because of their disabilities, dyslexia, or memorizing spells of words, thus, the next word suggestion has been developed to help them

1. **Expected Achievements**

The goal would be to accurately predict the next words in a sentence by considering the previous words from a given dataset. We could also collect our own emails and texts which we have written over the past years and then use that data to feed the algorithm. This would give us a clear picture of how this project works and will also make things easier for us.

1. **Organisation of Report**

Chapter 1: We will discuss the theoretical background behind Next Word Prediction, and we will examine its objective. We will also learn about the purpose, scope, and availability of the project.

Chapter 2: Survey of technologies, we are going to talk about the available technology that we could have used in our project. In addition, we'll discuss the technology we plan to use in our project and compare it to other technologies. Furthermore, we will talk about the domain of the project.

Chapter3: Requirement & Analysis chapter discusses the definition of the problem and the statements associated with the project, as well as the definition of the requirements, a feasibility plan, a schedule, and the most important conceptual model. The conceptual model explains what model we have used and why, as well as what its applications are. We will also see all the important diagrams related to the website.

Chapter 4: System Design, we discuss the models and features of Next Word Prediction. In addition, data integrity and constraint are discussed. Also, it tries to highlight security issues and user interfaces.

Chapter5: Implementation and Testing, Our implementation plan, standards, and protocols that were followed in implementing Next Word Prediction are described in this chapter. Our most significant point in this chapter is about the testing approach we have taken after the completion of the project. Here we discussed functional testing, non-functional testing, scalability and etc. Once the testing was complete, we note everything down as test cases.

Chapter6: Results and discussions, this chapter summarizes the test case we generated in the previous chapter as a test report. We also discuss the objectives behind the test cases. In addition, we drafted the user documentation for the user, as well as estimated the project's costs.

Chapter7: Conclusions, as the chapter name says in this chapter we talk about the conclusion of Next Word Prediction. We also mention the limitations and future scope of the project.

**CHAPTER 2 : SURVEY OF TECHNOLOGIES**

1. **Description of Available Technologies**

**N-gram models**: N-gram models are a type of language modeling approach in Natural Language Processing (NLP) that estimate the likelihood of a sequence of words**.**  N-gram models use a sliding window of n words to predict the next word in a sentence. This method is simple and efficient, but has limited accuracy compared to more complex models.

**Hidden Markov Models (HMMs):** HMMs are a probabilistic model that can be used for sequence prediction. They are often used in speech recognition and speech synthesis. Markov and Hidden Markov models are engineered to handle data which can be represented as ‘sequence’ of observations over time A lot of Machine Learning techniques are based on HMMs have been successfully applied to problems including speech recognition. A machine learning algorithm can apply Markov models to decision making processes regarding the prediction of an outcome.

**Long Short-Term Memory (LSTM) :** Recurrent neural network architecture designed to handle the problem of vanishing gradients in traditional RNNs. Excellent at handling long-term dependencies, can handle large amounts of training data. LSTMs can be computationally intensive, requires large amounts of data to train effectively

**Gated Recurrent Unit (GRU) :** Variation of the LSTM architecture that uses fewer parameters and is computationally more efficient. Computationally more efficient than LSTMs, still capable of handling long-term dependencies. May be less accurate than LSTMs for complex prediction tasks

1. **Comparative Analysis of Technologies in Chosen Area**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameters | N-gram models | Hidden Markov Models | LSTM | GRU |
| Architecture | Based on probability estimates of sequences of words or characters | Based on probabilistic state transitions | Recurrent neural network with memory cells and gates to control the flow of information | Simplified version of LSTM with two gates (update and reset) |
| Memory | Limited to N-gram context | Limited to Markov property | Has memory cells to store information and gates to control the flow of information | Has memory cells to store information and gates to control the flow of information, but with fewer parameters compared to LSTM |
| Handling long sequences | Struggles with long sequences | Struggles with long sequences | Can handle long sequences effectively | Can handle long sequences effectively |
| Computational complexity | Low | Moderate | High | Moderate |
| Speed | Fast | Moderate | Slow | Fast |
| Accuracy | Good for short sequences, struggles with longer sequences | Good for short sequences, struggles with longer sequences | Good for longer sequences, with the ability to capture long-term dependencies | Good for longer sequences, with the ability to capture long-term dependencies |

1. **Chosen Project Domain**

The domain I have chosen is Artificial Intelligence, Natural language processing (NLP) and Natural Language Understanding. NLP refers to the branch of computer science and more specifically, the branch of artificial intelligence or AI concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics rule-based modeling of human language with statistical, machine learning, and deep learning models. Natural Language Understanding is a branch of artificial intelligence. It is a subfield of [Natural Language Processing (NLP)](https://www.engati.com/glossary/neuro-linguistic-programming) and focuses on converting human language into machine-readable formats. Computers use NLU along with machine learning to analyze data in seconds. Working in this domain will help you learn many new things and increase your knowledge. This is the reason I chose Artificial Intelligence as my project domain.

1. **Technologies to be used**

Start to write from here

* 1. **Front End**

Streamlit

* 1. **Back End**

Keras Tokenizer

LSTM (Long-Short Term Memory)

GRU (Gated Recurrent Unit)

ModelCheckpoint

ReduceLROnPlateau

* 1. **Framework**

TensorFlow

* 1. **Other Development Tools**

Google Colab

1. **Reason Supporting the use of above selected technologies**

These technologies are very easy to use and the very appropriate in a machine learning algorithm. There are many documents that help us understand these technologies and provide reasons for everything. Colab allows anybody to write and execute arbitrary python code through the browser. TensorFlow provides pre-built functions and advanced operations to ease the task of building different neural network models. Keras Tokenizer is a text preprocessing tool used to tokenize text data into numerical representation in machine learning models. It's used to convert a text sequence into numerical values called tokens that can be easily processed by the model. LSTM is a special kind of recurrent neural network that are designed to handle the problem of vanishing gradients in traditional RNNs. GRUs are a variation of the LSTM architecture that uses fewer parameters and is computationally more efficient.

**CHAPTER 3 : REQUIREMENTS & ANALYSIS**

1. **Problem Statement and Problem Definition**

Give a statement for the problem you’re trying to solve

Define the problem statement in brief

1. **Requirements Specification**

What is requirement analysis?

The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. (write as it is)

* 1. **Functional Requirements**

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Functional requirements are as follows :

* Input and output capabilities: The model should be able to take a sequence of text as input, and output the probability distribution of the next word.
* Text processing: The model should be able to pre-process the input text to remove stop words, punctuations, and perform lower casing.
* Predictive accuracy: The model should be able to make accurate predictions of the next word, based on the input sequence.
* User interface: The model should have an easy-to-use interface for users to input the text and receive predictions.
* Model training: The model should be able to be trained on new data to improve its predictions.
* Model evaluation: The model should provide metrics to evaluate its performance, such as accuracy, precision, recall, and F1-score.
  1. **Non-functional Requirements**

Start to write from here

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior of functions.

Non-functional requirements are as follows :

* Compatibility

Compatibility would mean that the project is compatible with different operating systems, in different programming languages and on different devices.

* Reliability

Reliability would mean that the system accurately predicts the next word in a sequence, without fail, consistently and with a high degree of accuracy

* Availability

Availability refers to the amount of time a next word prediction project is functioning and accessible for use. The goal is for the project to be continuously available to users with minimal downtime for maintenance, upgrades, or failures.

* Security

In my project there isn’t really any need for security since we are not storing any data about our customers

* Accessibility

Accessibility would refer to the extent to which the project can be easily used and accessed by users, regardless of their abilities or disabilities

* 1. **User Requirements**

What is the user requirement?

User requirements are as follows :

* High accuracy in predicting the next word based on the input text
* Fast and responsive performance of the prediction engine
* Easy to use interface with minimal user input
* Compatibility with different platforms and devices
* Privacy and security of the input data
* The ability to customize predictions based on the user's preference
* A visually appealing and user-friendly interface
  1. **Hardware Requirements**

What are hardware requirements?

Hardware requirements are as follows :

* Laptop or Desktop
* CPU: Intel Core i3 or higher
* 8 GB RAM or more
  1. **Software Requirements**

What are software requirements?

Software requirements are as follows :

* Google Colab or Jupyter Notebook
* Installation of all the required libraries
* Pycharm or Visual Studio Code

1. **Feasibility**

A well-designed feasibility study should provide a historical background of the business or project,

a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations.

Generally, feasibility studies precede technical development and project implementation.

* 1. **Operational Feasibility**

It explains how well the proposed project solves the problem, how it takes into account the advantages and how it satisfies the requirements specified.

* Affordability

The goal of the next word prediction project is to provide accurate predictions at an affordable cost, so that it is accessible and usable by a wide range of people and organizations.

* Producibility

The goal of the next word prediction project may be to create a product that is both highly accurate and highly producible, so that it can be scaled and widely adopted in a timely and cost-effective manner

* Sustainability

The goal of the next word prediction project may be to create a product that is not only accurate and affordable, but also sustainable, so that it can be used effectively for many years to come. This may involve making the project scalable, easy to maintain, and resilient to changing conditions, such as changes in data, technology, or user needs

* 1. **Technical Feasibility**

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system.

* The goal of the next word prediction project may be to assess the technical feasibility of the project, in order to determine whether it is possible to implement the project within the desired time frame and budget, and whether it is likely to meet the required performance standards. This may involve conducting a thorough analysis of the technical requirements, constraints, and risks involved in the project, and evaluating the availability of the necessary technology and resources.
  1. **Economic Feasibility**

The purpose of an economic feasibility study (EFS) is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole.

* The goal of the next word prediction project may be to assess the economic feasibility of the project, in order to determine whether it is affordable and likely to provide a positive return on investment. This may involve conducting a thorough cost-benefit analysis, evaluating the potential benefits of the project in terms of increased efficiency, reduced costs, or improved user experience, and comparing these benefits to the costs involved in implementing and maintaining the project. The goal is to determine whether the project is economically feasible, meaning that the benefits of the project are likely to outweigh the costs over the long term

1. **Planning and Scheduling**

What is planning?

Planning a next word prediction project involves several key steps, including defining the project objectives, conducting a feasibility study, determining the resources required, and creating a detailed project plan.

What is scheduling?

Scheduling a next word prediction project involves creating a detailed project timeline that includes all the major tasks, activities, and milestones involved in the project.

* 1. **Gantt Chart**

Insert the gantt chart diagram here

Prepare a gantt chart preferably with the help of excel based

* 1. **Pert Chart**

Insert the pert chart diagram here

Prepare the pert chart as was discussed in the lecture

1. **Preliminary Product Description**

Preliminary product description helps in identifying the requirements and the objectives of the new proposed product/project/system. It helps in defining the functions and associated activities or operations of the proposed product/project/system.

* 1. **Short summary of the objective (do not include this header)**

A preliminary product description of a next word prediction project would include an overview of the product's key features and functions, as well as its target audience and intended use cases. The description might also include information about the technology used to develop the product, such as machine learning algorithms or natural language processing techniques.

* 1. **Short summary of the requirements (do not include this header)**

Some specific information that might be included in the preliminary product description for a next word prediction project are:

Objectives: A clear and concise statement of what the product is intended to do, and the specific problems or challenges it is designed to address.

Target audience: A description of the user base for the product, including demographic information, technical expertise, and other relevant factors.

Key features: A list of the most important features and capabilities of the product, and how they meet the needs of the target audience.

Technical requirements: An overview of the technical infrastructure and resources required to implement and use the product, including hardware, software, and other technology.

Performance standards: A description of the performance criteria that the product is expected to meet, such as accuracy, speed, and scalability.

Cost and pricing: An estimate of the total cost of the project, including the cost of development, deployment, and maintenance, as well as information about pricing and revenue models.

1. **Conceptual Model**
2. **Process Model**

Process models are [processes](https://en.wiktionary.org/wiki/Process) of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. One possible use of a process model is to prescribe how things must/should/could be done in contrast to the process itself which is really what happens (write this as it is)

**Proposed Process Model**

“For IOT oriented project, prepare complete flowchart about working from start to the end”

* **Name of process model that fits your project**

AGILE MODEL

* **Brief overview of the process model (short paragraph)**

Agile is a project management methodology that emphasizes flexibility and collaboration in the development process. It is based on iterative development, with a focus on delivering small, incremental improvements to a product or service over time. The Agile approach values the following:

Individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation.

Responding to change over following a plan.

* **Design of the process model (diagram of the chosen model)**



* **Reasons for choosing this process model**

Reasons for choosing an Agile model include:

Faster Time to Market: Agile allows for quick delivery of working software, which can be delivered to customers for testing and feedback, resulting in a faster time to market.

Increased Flexibility: Agile accommodates changing requirements and allows teams to respond quickly to change, making it easier to adapt to new challenges or opportunities.

Improved Collaboration: Agile encourages close collaboration between team members, stakeholders, and customers, resulting in better communication and more effective problem-solving.

Better Quality: Agile emphasizes the delivery of working software, allowing teams to focus on testing and quality assurance, resulting in higher-quality products.

Increased Visibility: Agile provides greater visibility into the progress of a project, allowing teams to identify and resolve problems quickly.

Enhanced Customer Satisfaction: Agile allows for more frequent interaction with customers, enabling teams to gather feedback and make improvements based on that feedback, resulting in greater customer satisfaction.

While Agile is not the right choice for every project, its focus on flexibility, collaboration, and customer satisfaction makes it a popular choice for many organizations.

* **Application of chosen process model**

Agile is a popular project management methodology that is widely used across a variety of industries and applications. Some of the common applications of the Agile model include:

Software Development: Agile is particularly well-suited to software development projects, where requirements can change frequently and unpredictably. It allows teams to quickly adapt to changes and deliver working software in short iterations.

IT Projects: Agile can be applied to a variety of IT projects, including infrastructure, network, and security projects, where requirements and technology can evolve rapidly.

Marketing and Advertising: Agile can be used in marketing and advertising projects to quickly test and refine ideas, improving the chances of success.

Product Development: Agile is often used in product development projects, where teams can test and refine prototypes in short iterations, resulting in improved product design and functionality.

Healthcare: Agile can be used in healthcare projects to quickly adapt to changes in regulations, technologies, and patient needs, resulting in better patient outcomes.

Financial Services: Agile can be applied in financial services projects, where regulatory and market requirements can change rapidly, allowing teams to respond quickly and effectively.

These are some of the common applications of the Agile model, but the methodology can be applied in many other industries and projects where flexibility, collaboration, and quick adaptation to change are valued.

* **Advantages of chosen process model**

Flexibility: Agile allows for changes in requirements and priorities, making it easier for teams to respond quickly to new challenges and opportunities.

Improved Collaboration: Agile encourages close collaboration between team members, stakeholders, and customers, resulting in better communication and more effective problem-solving.

Faster Time to Market: Agile allows for the quick delivery of working software, which can be delivered to customers for testing and feedback, resulting in a faster time to market.

Increased Visibility: Agile provides greater visibility into the progress of a project, allowing teams to identify and resolve problems quickly.

Enhanced Customer Satisfaction: Agile allows for more frequent interaction with customers, enabling teams to gather feedback and make improvements based on that feedback, resulting in greater customer satisfaction.

Better Quality: Agile emphasizes the delivery of working software, allowing teams to focus on testing and quality assurance, resulting in higher-quality products

* **Disadvantages of chosen process model**

Lack of Detailed Planning: Agile emphasizes flexibility and adaptation, which can result in a lack of detailed planning and foresight, making it more difficult to manage resources and budgets.

Difficulty with Predicting Outcomes: Agile prioritizes flexibility over predictability, making it more difficult to accurately predict project outcomes, such as cost, schedule, and scope.

Challenges with Integration: Agile projects are often composed of many small, loosely coupled pieces, making it more difficult to integrate these pieces into a cohesive whole.

Complexity: Agile can introduce complexity into a project, making it more difficult to manage and understand.

Difficulty with Compliance: Agile can be difficult to align with regulatory requirements and other constraints, making it more challenging to meet compliance needs.

1. **The goals of a process model are to be:**
2. Descriptive

Track what actually happens during a process

1. Prescriptive

Define the desired processes and how they should/could/might be performed.

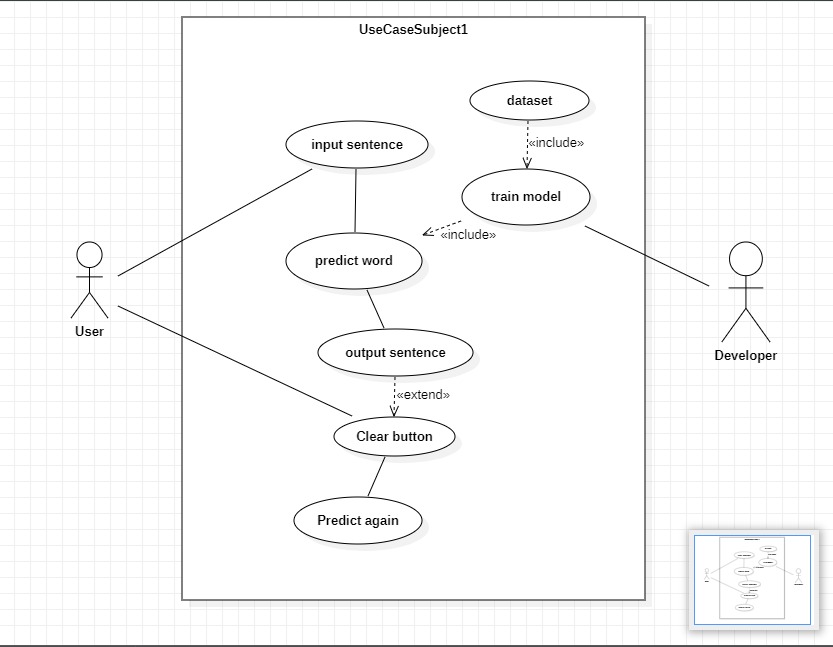
1. Explanatory

Provide explanations about the rationale of processes.

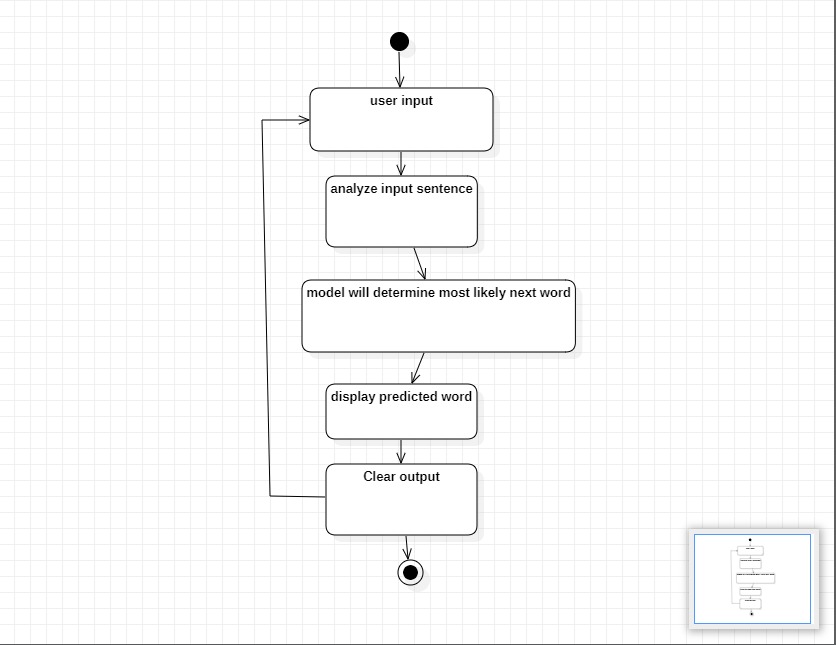
1. **Diagrams to be included in the design phase are as follows:**

For IOT oriented projects, everything needs to be drawn except ER diagram

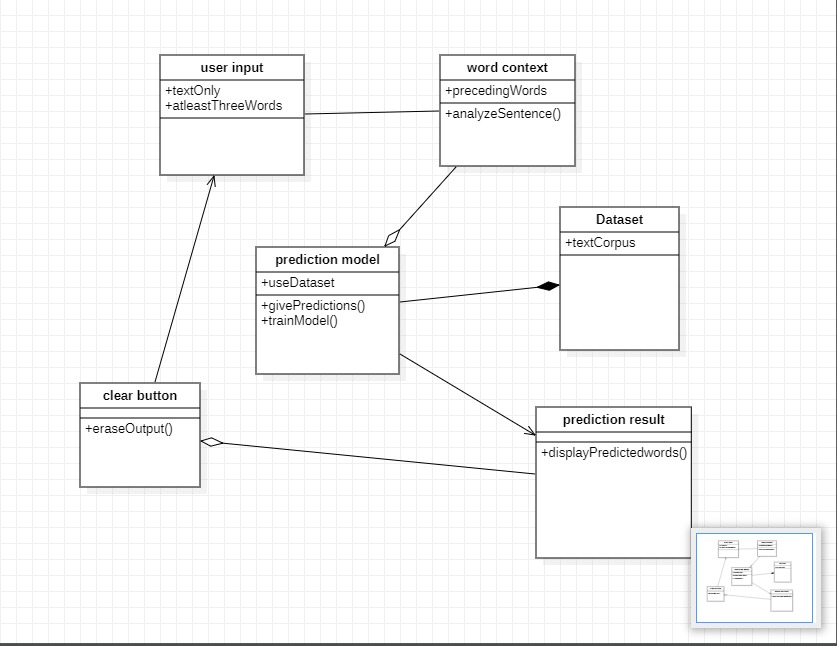
1. **Use case diagram**

****

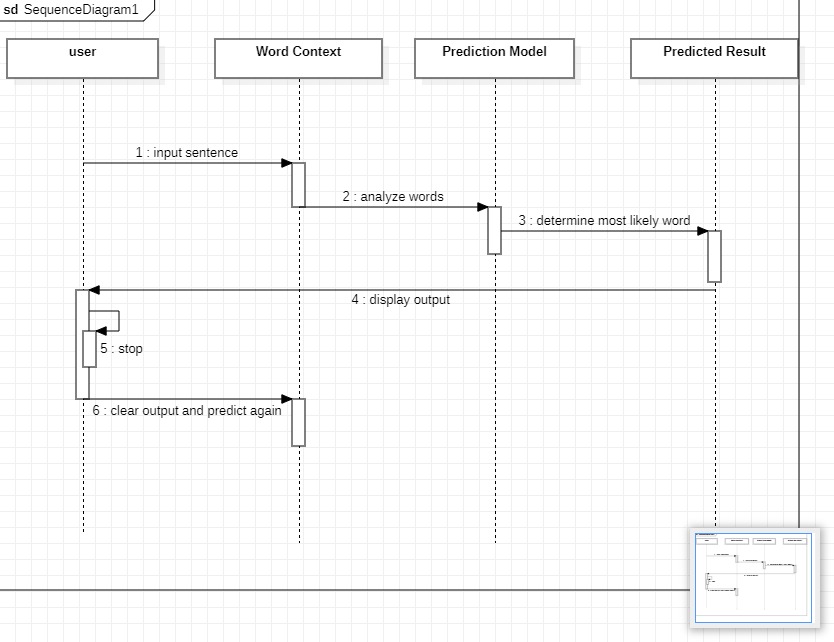
1. **Activity diagram**

****

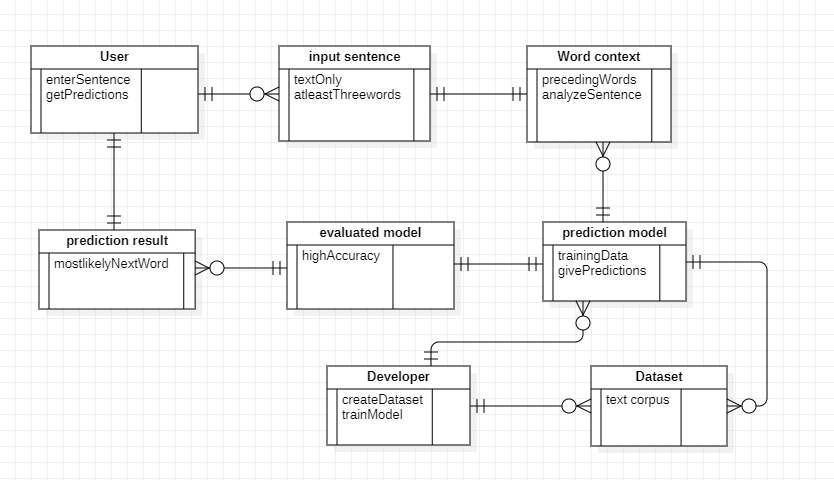
1. **Class diagram**

****

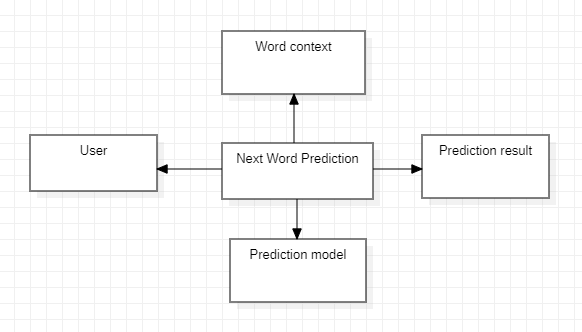
1. **Sequence diagram**

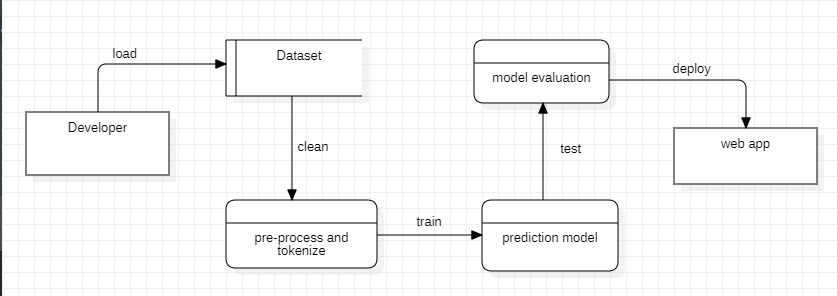
****

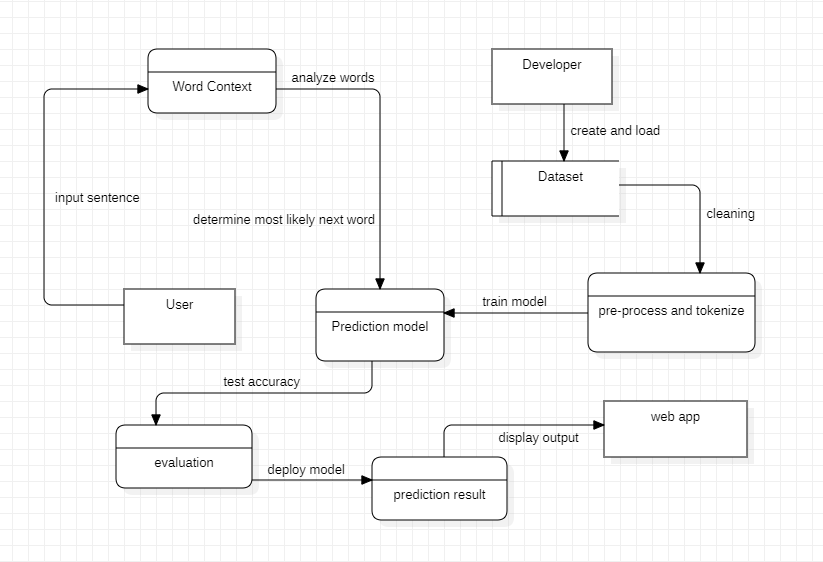
1. **E-R model**

****

1. **Data Flow Diagram**

****

****

****

**CHAPTER 4 : SYSTEM DESIGN**

1. **Basic Modules**

Modules represent your web app or project. You can create individual modules or combine modules to make it more effective

* 1. **Description of Desired Modules**

Start to write from here

The desired modules in a next word prediction project depend on the specific requirements and goals of the project. However, here are some common modules that are typically included in a next word prediction project:

Data Collection: This module is responsible for collecting and preprocessing the text data that will be used to train the prediction model. It may include functions for cleaning, tokenizing, and stemming the text data.

Model Training: This module trains a machine learning model on the preprocessed text data to predict the next word based on the context of the current word. It may use algorithms such as recurrent neural networks (RNNs), transformer models, or other deep learning techniques.

Model Deployment: This module integrates the trained model into a production environment, such as a web application or mobile app, making the predictions available to end-users.

Prediction Engine: This module is responsible for making predictions based on the context of the current word. It may include functions for evaluating the predictions and choosing the most likely next word.

User Interface: This module provides a user-friendly interface for end-users to interact with the next word prediction system. It may include functions for displaying the predicted next word and allowing the user to correct or modify the prediction.

Performance Monitoring: This module monitors and evaluates the performance of the next word prediction system over time. It may include functions for tracking accuracy, speed, and scalability, and for making improvements to the model as necessary.

* 1. **Description of Desired Features**

The desired features in a next word prediction project will depend on the specific goals and requirements of the project. However, some common features that are typically desired in next word prediction projects include:

* Contextual Awareness
* Accuracy
* Speed
* Flexibility
* Scalability
* Integrity
* User Experience
* Privacy

These features can be used as a starting point for defining the requirements and goals of a next word prediction project. It's important to prioritize and balance the features based on the specific requirements of the project to ensure that the final solution meets the needs of the users and stakeholders.

1. **Data Design**

In the design phase, the requirements will be broken down further to be able to forecast the project’s timeline and estimate the level of effort and amount of resources needed. Design is a very important phase and is a multi-step process which represents structure, program, interface characteristics and procedural details. The proposed system is designed using the design models such as functional decomposition diagrams, data flow diagrams, entity relationship diagrams or any unified modeling language diagrams. The design phase includes all the diagrams which provide an outline of how the application would look.

1. Schema Design
   * Explain the structure of the database

The schema design of a text corpus dataset refers to the structure and organization of the data stored in the dataset. The schema defines the structure of the data, including the types of data fields, the relationships between fields, and the constraints on the data.

The schema design of a text corpus dataset can vary depending on the specific needs of the project and the data being analyzed. The design should be carefully planned to ensure that the data is organized in a way that supports the goals of the project and makes it easy to access and analyze the data.

1. Data Integrity and Constraints
   * Integrity

Data integrity refers to the accuracy and completeness of the data in a dataset. In the context of a text corpus dataset, data integrity is concerned with ensuring that the text data is correctly recorded and stored in the dataset.

* + Constraints

Constraints are rules that are placed on the data in a dataset to ensure data integrity

Data Type Constraints: These ensure that the data is stored in the correct format, such as text or integer.

Data Range Constraints: These ensure that the data falls within a specified range.

Data Uniqueness Constraints: These ensure that there are no duplicates in the data.

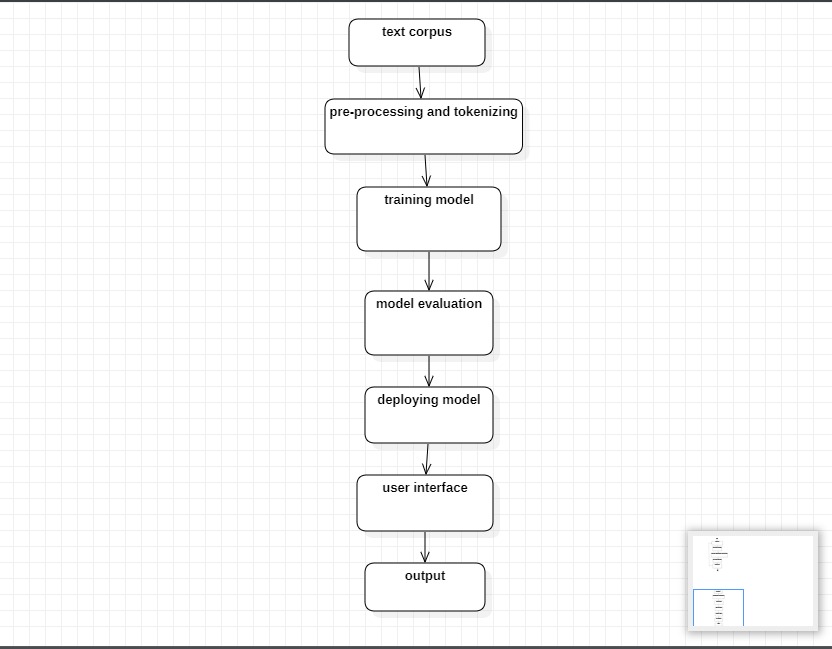
Data Referential Integrity Constraints: These ensure that relationships between data in the dataset are maintained.

These constraints help ensure the accuracy and completeness of the data in the text corpus dataset, and they play an important role in maintaining data integrity in the dataset. By enforcing these rules, the dataset becomes more reliable and trustworthy, and it is easier to identify and correct errors in the data.

1. **Procedural Design**
2. Logic Diagram
   * **Define the systematic flow of procedure (with programmers point of view)**

First we have to download a text corpus which we can use as a dataset for our model. Once we get a good corpus we have to pre-process/clean the data. This includes getting rid of all punctuations and non-text characters. Next we tokenize our data, which basically means that we assign a unique index to every word in our dataset. Then we create our lstm/gru model. The cleaned dataset is used to train our model. The model will learn the relationship between words in the corpus and use that information to predict words. Once model is trained it will be evaluated to check its accuracy and efficiency. Finally the model is deployed on a web app. The user can then enter a sentence and get predictions form the model. The predicted sentence will be displayed to the user

* + **Process Flow diagram or Control flow diagram or Circuit diagram(For IOT)**

****

1. Data Structures
   * **Define the data structure that you have planned in all your process**

In machine learning (ML) models, data structures are used to store and organize the data used to train and test the models

Lists: An ordered collection of elements, where each element can be of different types and lengths. Lists can be used to store sequences of data, such as sequences of words in a sentence.

1. Algorithm design
   * **Explain the working of the algorithm**
   * **Give proper explanation regarding input, processing, core logic and output**

RNNs are popular for language modeling as they have a memory mechanism that allows them to take into account the context of the previous words in a sequence when making predictions. The RNN processes the input sequence one word at a time, and its hidden state is updated with each word to capture the context of the previous words. The final hidden state is then used to make a prediction for the next word in the sequence.

LSTM and GRU are both types of RNNs that are designed to handle the vanishing gradient problem in traditional RNNs, where the influence of earlier inputs on the hidden state decreases as the sequence becomes longer. Both LSTM and GRU have gating mechanisms that allow them to control the flow of information through the network and maintain information for longer periods of time.

LSTMs have three gates: input, forget, and output gates, which control the flow of information into and out of the cell state. GRUs have two gates: a reset gate, which decides how much of the previous hidden state to discard, and an update gate, which decides how much of the previous hidden state to keep.

In a next word prediction project, the LSTM or GRU model is trained on a large text corpus to predict the next word in a sequence given the previous words. The model takes the input sequence one word at a time, updates its hidden state based on the input and previous hidden state, and uses the final hidden state to make a prediction for the next word.

The choice of LSTM or GRU for a specific project will depend on various factors, such as the size of the text corpus, the computational resources available, and the desired level of accuracy.

* + **Frame the algorithm in the forms of steps**

Preprocessing: Clean and preprocess the text data to prepare it for modeling. This may include converting the text to lowercase, removing punctuation, stemming/lemmatizing words, etc.

Tokenization: Convert the text into sequences of tokens (words or subwords) that can be fed into the model.

Embedding: Convert the tokens into numerical representations that can be used as input to the model. This is typically done using word embeddings, which are dense vectors that capture the semantic meaning of words.

Define the model architecture: Choose a LSTM or GRU model architecture and define the network structure, including the number of layers, the number of units in each layer, the activation function, etc.

Training: Train the model on the preprocessed text data by minimizing the cross-entropy loss between the predicted next word and the actual next word. The model is trained using backpropagation and optimization algorithms such as SGD or Adam.

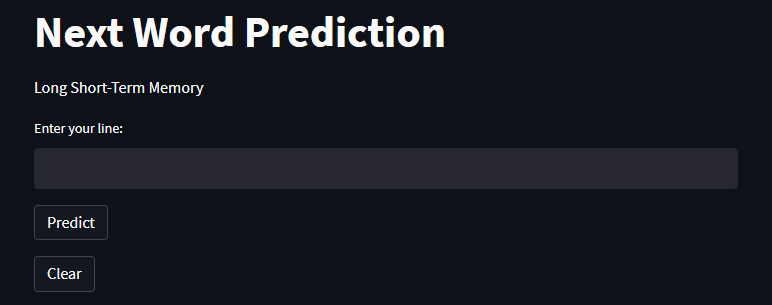
Testing: Evaluate the model on a held-out test set to determine its accuracy. Common metrics for next word prediction include perplexity and prediction accuracy.

Deployment: Deploy the trained model in a real-world application, such as a keyboard app that predicts the next word as the user types.

1. **User Interface Design**
2. Define the user task and environments availability to carry that task

The user task would be to input a sentence which contains atleast three words for the model to predict accurately. The sequence of words used as input should be from the dataset on which the model was trained. Then the user has to click on the predict button. The model will then start considering the previous words of the input sentence and give a prediction. If the user wants the model to predict again he/she can simply hit the clear button which clears the output and can type in a sentence again for the model to give a prediction.

1. Describe internal and external components of the architecture or user interface
2. Draw or frame sample user interface design



1. **Security Issues**
2. **List and describe all the security issues that might be experienced while a user uses your application.**

There are not any major security issues that would occur in my project. This project is just an algorithm that would predict the next words in a sentence and give you possible suggestions by understanding your patterns of writing/texting. One thing we could be careful of is when we are training the model using our own data, which could be either emails or texts, we must make sure this data does not get leaked. The data must be kept confidential

1. **Give your plan of action on how would you treat those issues**

The most common way to keep your data secure is to encrypt it. Modern tools make it possible for anyone to encrypt emails and other information.

Another way to keep your data secure is by adding a firewall. Firewalls assist in blocking dangerous programs, viruses or spyware before they infiltrate your system. Various software companies offer firewall protection, but hardware-based firewalls, like those frequently built into network routers, provide a better level of security.

1. **Dataset - only for AI oriented projects**

Study the data you are planning to use for the project

● Name of the dataset:

Human Conversations training Data

Harry Potter Books Corpora

Seafood Recipes

● Source of the dataset:

<https://www.kaggle.com/datasets/projjal1/human-conversation-training-data>

<https://www.kaggle.com/datasets/balabaskar/harry-potter-books-corpora-part-1-7>

<https://www.pdfdrive.com/seafood-recipes-d34780763.html>

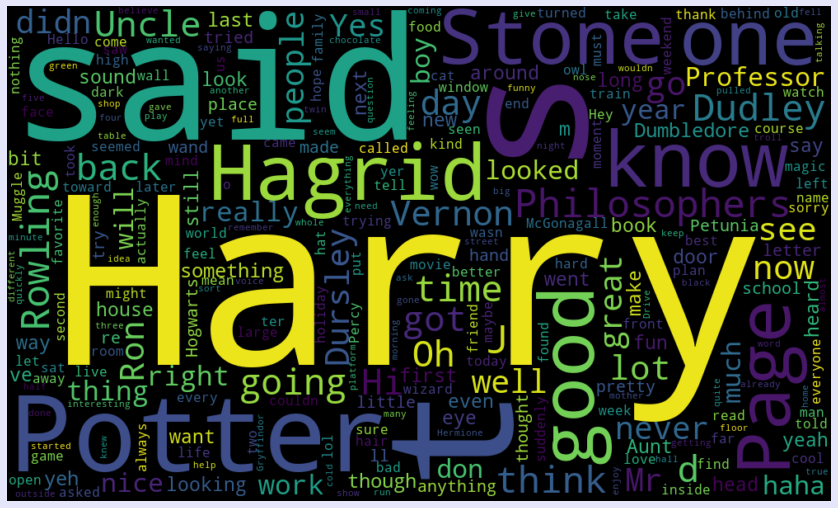
● Size of the dataset:

99,394 words, 566 kB

* Number and list of features present in the dataset :

Plain Text Corpus UTF-8

* Perform exploratory analysis of the dataset
  1. **Wordcloud:** A word cloud is a collection, or cluster, of words depicted in different sizes. The bigger and bolder the word appears, the more often it’s mentioned within a given text and the more important it is. Word clouds are great for visualizing unstructured text data and getting insights on trends and patterns.



* 1. **Lemmatization and Stemming**

Lemmatization: The process of reducing the different forms of a word to one single form. The purpose of lemmatization is same as that of stemming but overcomes the drawbacks of stemming. In stemming, for some words, it may not give may not give meaningful representation. Here, lemmatization comes into picture as it gives meaningful word.

Result after Lemmatization

['If could go anywhere vacation , would go ?', 'I like rainforest , I know requires extensive training beforehand I heard rainforest southeast Asia zipline tree tree .', 'When younger , Harry dreamed unknown relation coming take away , never happened ; Dursleys family .']

Stemming: Stemming is a natural language processing technique that is used to reduce words to their base form, also known as the root form. The process of stemming is used to normalize text and make it easier to process. Stemming is a process that stems or removes last few characters from a word, often leading to incorrect meanings and spelling.

Result after Stemming

['if could go anywher vacat , would go ?', 'i like rainforest , i know requir extens train beforehand i heard rainforest southeast asia ziplin tree tree .', 'when younger , harri dream unknown relat come take away , never happen ; dursley famili .']

* 1. **Bag of Words**

A bag of words is a representation of text that describes the occurrence of words within a document. We just keep track of word counts and disregard the grammatical details and the word order. One of the biggest problems with text is that it is messy and unstructured, and machine learning algorithms prefer structured, well defined fixed-length inputs and by using the Bag-of-Words technique we can convert variable-length texts into a fixed-length vector.

text1 = ['Human Conversation training data',

'Harry Potter and the Philosophers Stone']

Results

['conversation' 'data' 'harry' 'human' 'philosophers' 'potter' 'stone'

'training']

Human Conversation training data

[1 1 0 1 0 0 0 1]

Harry Potter and the Philosophers Stone

[0 0 1 0 1 1 1 0]

**CHAPTER 5 : IMPLEMENTATION AND TESTING**

1. **Implementation Approaches**

The implementation approach/plan would be to first collect a large corpus of text data that will be used to train the model. Then the cleaning and preprocessing of the data to prepare it for modeling. After data is cleaned it should be tokenized which means converting the text into sequences of tokens. The tokens are then converted into numerical representations that can be used as input to the model. We have to first create the model architecture and define some parameters of the model. Then we have to train the model on the preprocessed text data. After model has trained we will evaluate it to determine its accuracy. Finally we will deploy the trained model in our web application.

* 1. **State the standards and protocols used in implementation**

In the implementation of a next word prediction project, several standards and protocols are commonly used to ensure the compatibility and robustness of the model.

Data Format: The text data used to train the model should be in a standard format, such as plain text or Unicode.

File System: A standard file system should be used to store the data and model to ensure reliability and accessibility.

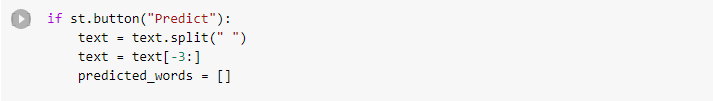
Model Format: The model should be stored in a standard format that can be easily loaded and used in different systems.

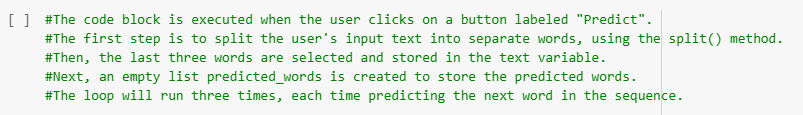
Model API: A standard model API should be used to serve the model and provide an interface for making predictions.

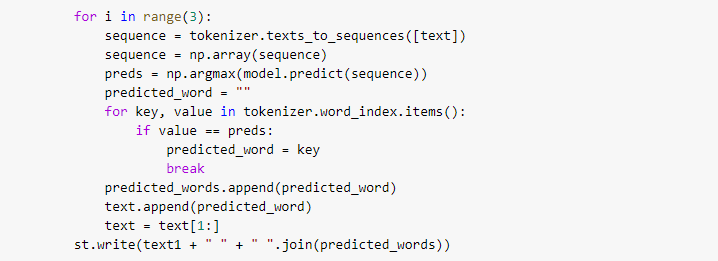
1. **Coding Details and Code Efficiency**

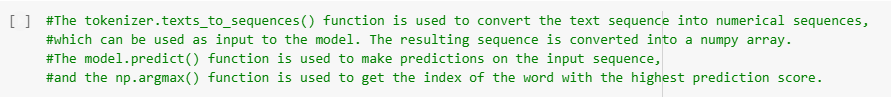
The backend code includes preprocessing the dataset, tokenizing and creating and training the model. The trained model is then saved and integrated to my frontend code. I have used TensorFlow and Keras for my backend and Streamlit for my frontend code.

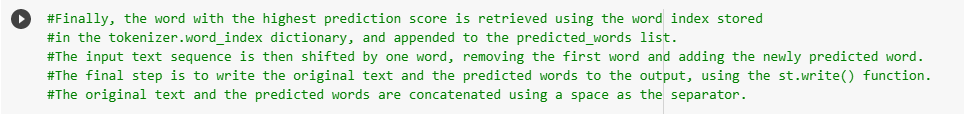
* 1. **Code of the main logic (must be with comments)**



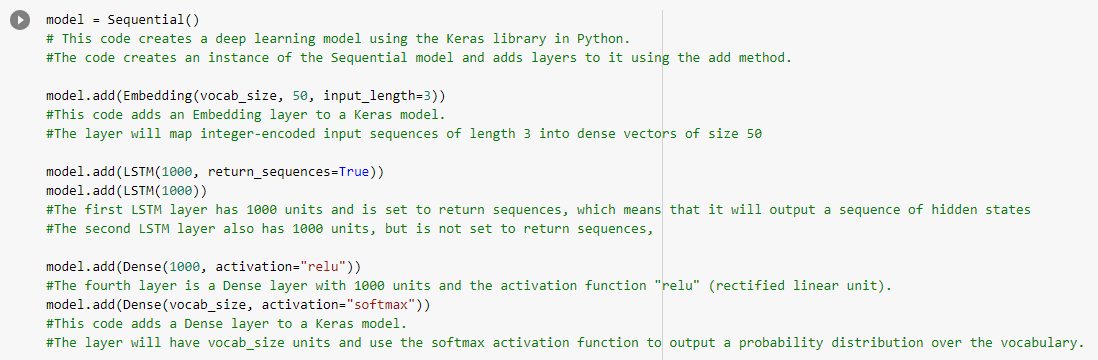


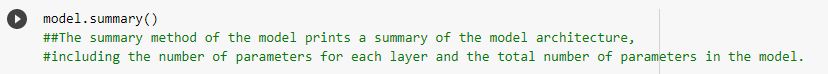


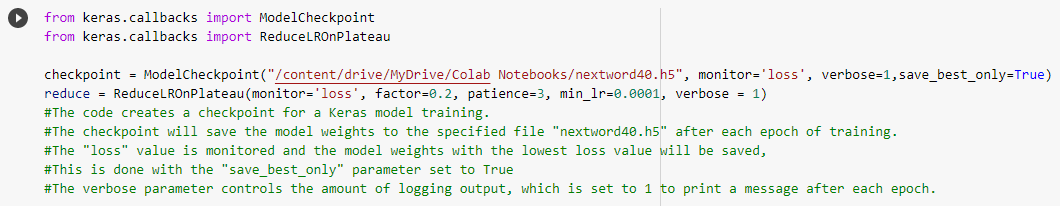


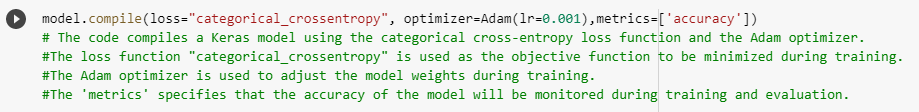


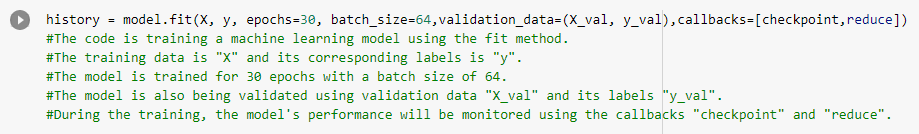
* 1. **Code of the algorithm**

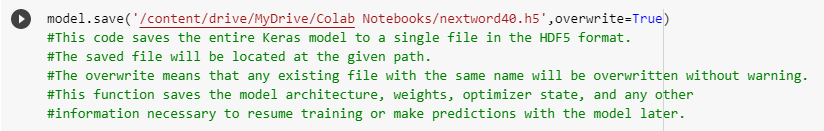
****

****

****

****

****

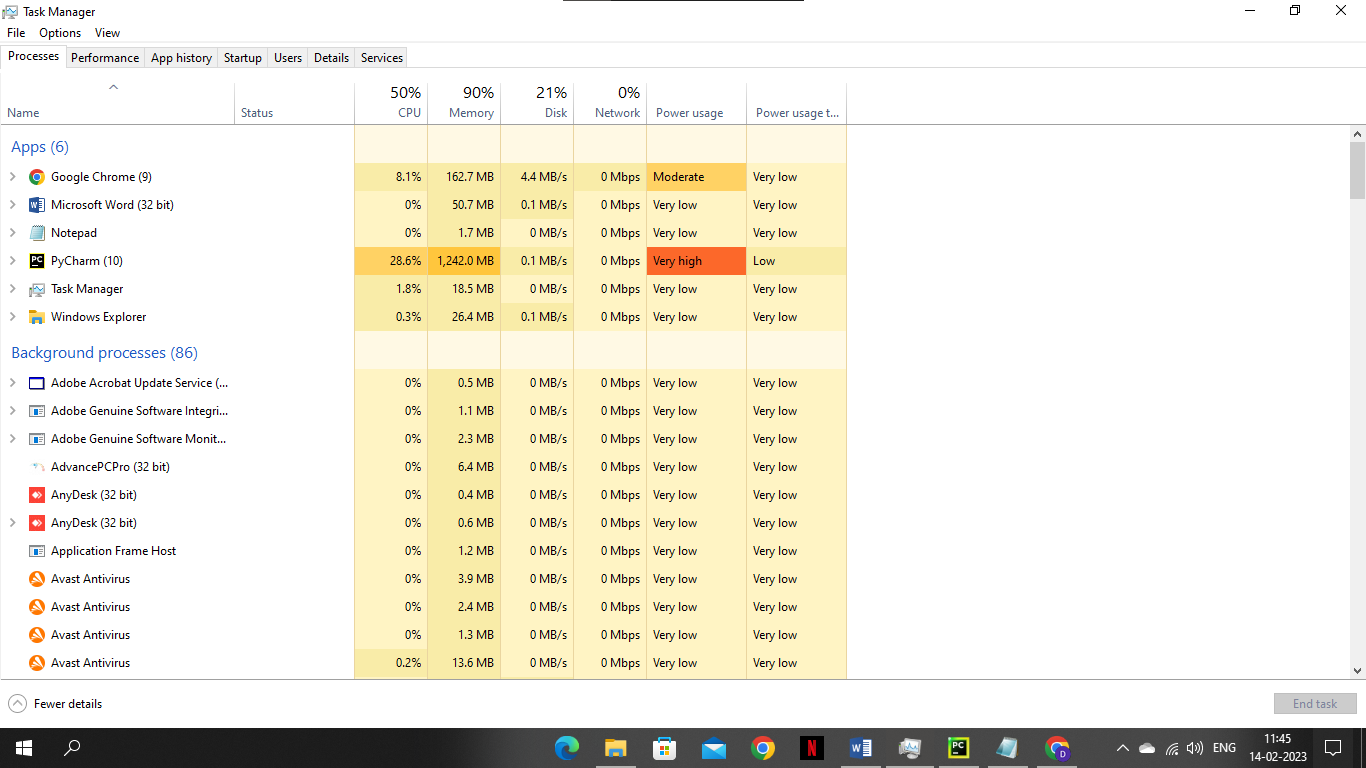
****

* 1. **Code Efficiency**

Explain how effective and efficient your code is?

The code for my project is effective because it solves the problem it was designed for. It is able to give predictions for a sequence of words as input. My model has an accuracy of 90.0%. I have also tested my model to see if it works well on a variety of input data.

Since it is a machine learning model, creating and training the model requires high computing power and system resources. The model takes a few minutes to give you predictions once you input a sentence.



1. **Testing Approach**

Define and explain a scheme for testing.

Keep into consideration that the approach used for testing is dependent on the system design that you have specified.

The testing approach for a next word prediction project would typically involve a combination of functional testing and non-functional testing, with a focus on verifying the accuracy and performance of the LSTM and GRU models.

* 1. **Functional Testing**

Functional testing is a type of software testing that focuses on evaluating the functional requirements and specifications of an application. In the context of a next word prediction project, functional testing would involve evaluating the ability of the LSTM and GRU models to perform the specific functions that they were designed to perform, such as predicting the next word in a sentence and that they are functioning as intended, without any issues or bugs.

1. **User Acceptance Testing or Beta Testing**

Beta testing is a type of testing that is performed on a software product in the final stages of development before its official release. In the case of a next word prediction project, beta testing would involve evaluating the performance and accuracy of the LSTM and GRU models on a sample of real-world data. The goal of beta testing is to identify any remaining bugs or issues with the models.

User acceptance testing (UAT) is a type of testing that is performed to determine if the next word prediction project meets the requirements and expectations of the end users. UAT is typically performed after the beta testing phase and before the final release of the product.

1. **Unit Testing**

Unit testing is a software testing method that involves testing individual units or components of a software application to ensure that they are working correctly. In the context of a next word prediction project, unit testing would involve testing the individual components of the LSTM and GRU models to verify that they are functioning as expected.

Unit testing is typically performed by the development team as part of the software development process

1. **Integration Testing**

Integration testing is a type of software testing that involves testing the interactions between the different components of a software application. In the context of a next word prediction project, integration testing would involve testing the interactions between the LSTM and GRU models and any other systems or components that they interact with.

The goal of integration testing is to identify and resolve any issues that arise when different components of the application are combined

* 1. **Non-Functional Testing**

Non-functional testing is a type of software testing that focuses on evaluating the non-functional characteristics of an application. Non-functional testing includes a wide range of testing techniques that are used to evaluate the quality attributes of an application, such as its performance, scalability, security, usability, and more.

In the context of a next word prediction project, non-functional testing could involve evaluating the performance of the LSTM and GRU models under different conditions, such as when handling large volumes of data, when dealing with complex and long sentences, or when being used in real-time applications.

The goal of non-functional testing is to evaluate the overall quality of the models and to ensure that they meet the requirements for performance, reliability, and usability

1. **Performance Testing**

Performance testing is a type of software testing that involves evaluating the performance of a software application under different conditions, including load, stress, and capacity. In the context of a next word prediction project, performance testing would involve evaluating the performance of the LSTM and GRU models, and determining how they perform under different conditions.

The goal of performance testing is to measure the response times and resource utilization of the models, and determine if the models are scalable and can handle the expected load and usage

1. **Scalability Testing**

Scalability testing is a type of performance testing that involves evaluating the ability of a software application to handle increasing load or usage over time. In the context of a next word prediction project, scalability testing would involve evaluating the ability of the LSTM and GRU models to handle increasing amounts of data or usage as the product grows and becomes more popular.

The goal of scalability testing is to determine the maximum load or usage that the models can handle, and to identify any performance bottlenecks that may limit the ability of the product to scale

1. **Portability Testing**

Portability testing is a type of testing that assesses the ability of a software application to be easily moved or adapted to different platforms, operating systems, or hardware environments. In the context of a next word prediction project, portability testing would involve evaluating the ability of the LSTM and GRU models to run on different platforms, operating systems, or hardware configurations, and to be easily adapted to new environments as needed.

The goal of portability testing is to identify any compatibility issues that may need to be addressed in order to make the models more portable.

* 1. **Black Box Testing**

Blackbox testing is a type of software testing that focuses on the functionality of an application without considering the internal implementation or structure. In the context of a next word prediction project, blackbox testing would involve evaluating the functionality of the LSTM and GRU models from the perspective of a user, without considering the underlying algorithms or code that powers the models.

The goal of blackbox testing is to ensure that the models work as expected and meet the requirements of the end-user

* 1. **White Box Testing**

Whitebox testing is a type of software testing that focuses on the internal implementation and structure of an application. In the context of a next word prediction project, whitebox testing would involve evaluating the algorithms and code that power the LSTM and GRU models, with a focus on verifying the correctness of the implementation and identifying any bugs or issues.

The goal of whitebox testing is to ensure that the models are functioning as intended, and to identify any potential weaknesses or areas for improvement in the underlying algorithms and code.

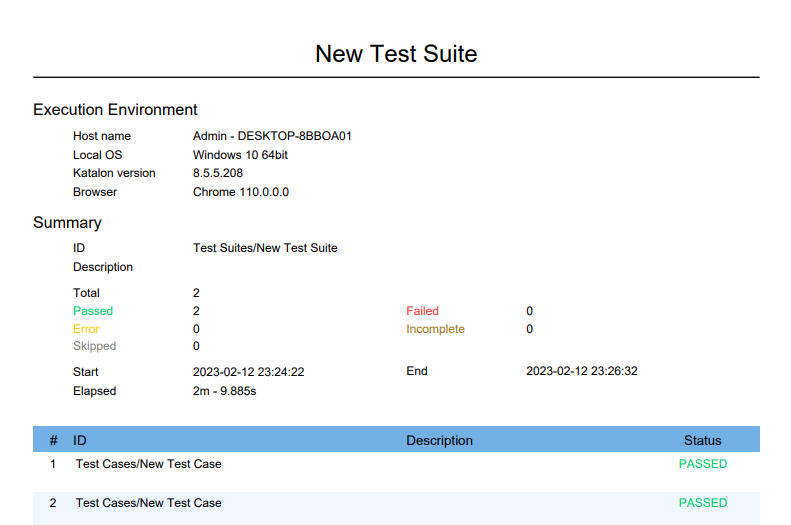
1. **Test Cases**

Prepare a test case for each module of your project covering all the functionalities and testing every component within it.

Prepare test cases in one excel workbook and use a different sheet for each module’s test case.

Follow the format given in the excel sheet for writing test cases 🡪

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test ID | Test Case Description | Test Case Procedure | Expected Output | Actual Output | Date | Result | Note, if any |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

****

1. **Modification and Expected Improvements**

Once testing is done and if there are any bugs or errors, modification needs to be made.

Make changes in your code and explain what modifications you had to do to improve your code here.

After I initially trained my model and tested it, the model gave very random and inaccurate predictions. To improve my model efficiency I had to increase the number of layers of my model. I had to increase my epochs too and keep training my model continuously until I could get a low loss and high accuracy.

The second change I did in my code was to predict three words instead of just one word. The reason was that when the model predicted just one word, the sentence usually came out to be very meaningless/incomplete. Predicting three words gave a meaning to your sentence and looked more complete too.

**CHAPTER 6 : RESULTS AND DISCUSSIONS**

1. **Test Reports**

Prepare test report in the excel sheet as given on the google classroom

Paste the excel sheet test report here

For the whole project prepare one test report which covers the following points

* Project Name
* Test Objective
  + Write objective of each stage of testing i.e. testing for each module and also functional, non functional testing
* Test Summary
  + Take this from excel sheet but frame it in the form of sentences

1. **User Documentation**

Prepare a user manual from start to end of your project with proper and complete details for better understanding of working and functionality of your project

User Manual for Next Word Prediction Project

Introduction:

This user manual provides instructions on how to use the next word prediction project. The project is a machine learning model that predicts the next word in a sentence based on the input text. The model is trained on a large corpus of text and is designed to provide accurate and relevant predictions.

Installation: To use the next word prediction project, you must first install the necessary software and dependencies. This will include the required programming language, such as Python, as well as any relevant libraries, such as TensorFlow.

* Input text: The next word prediction model takes as input a sentence or partial sentence. The input text should be in the form of a string.
* Model selection: The project includes both LSTM and GRU models, and the user can choose which model to use for prediction. The choice of model will depend on the specific requirements of the task and the desired level of accuracy.
* Model prediction: The model will return the most likely next words based on the input text. The prediction will be presented as a string.
* Real-time prediction: The model is designed to provide real-time predictions, so the prediction will be made immediately after the input text is provided.

Conclusion:

The next word prediction project is a powerful tool for predicting the next word in a sentence. With its real-time predictions and ability to be fine-tuned for specific domains, it is an ideal solution for a wide range of applications. Whether you are a researcher, developer, or end-user, the next word prediction project is a valuable resource for understanding and using language more effectively.

1. **Cost Estimation**

Cost estimation models are mathematical algorithms or parametric equations used to estimate the costs of a product or project. The results of the models are typically necessary to obtain approval to proceed, and are factored into business plans, budgets, and other financial planning and tracking mechanisms.

* **The Development Model**

COCOMO (Constructive Cost Model) is a regression model based on LOC viz. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

* **Key Parameter**

1. Efforts - measured in person months units
2. Schedule - measured in span of months or weeks

To estimate the effort and development time, COCOMO uses the same equations but with different coefficients (a, b, c, d in the effort and schedule equations) for each development mode. Types are as follows :

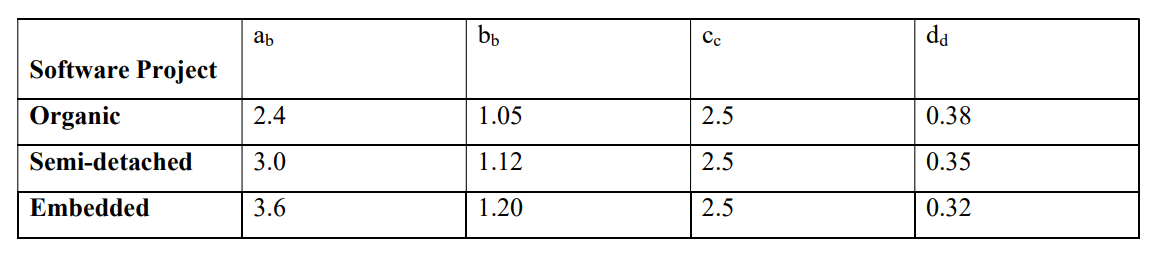
* Organic System
* Semi - detached System
* Embedded System

The basic COCOMO equations take the form

* Effort Applied (E) = ab (KLOC,0.3) bb [person-months]
* Development Time (D) = cb (Effort Applied) db [months]
* People Required (P) = Effort Applied / Development time [count]

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for a project.

The coefficient ab, bb, cb and db are given in the following table:



**COCOMO Model for “Next Word Prediction”**

1. Effort : 0.7
2. Time for development : 1.23

Where,

Effort = Number of staff months (SM)

Size = Number of source lines of code

Time = Total number of months required to complete the project

The Project Code for Next Word Prediction application contains 300 Lines of code

Since, we know that 1000 Lines of Code = 1 KLOC (K - Kilo - 10^3)

Therefore, the project consists of 0.3 KLOC.

Effort = = 0.7 SM

Time for development = = 1.23 Months

Cost per Month = Rs.1000/-

Total Cost of the Project = Cost per Month \* Time required for the development project

= 1000 \* 1.23

= Rs. 1230

**CHAPTER 7 : CONCLUSIONS**

1. **Conclusion**

Final outcome of the project you have prepared.

The conclusion of your project must be at least 1 page, segregated into 2 or 3 paragraphs

Top of Form

In conclusion, the next word prediction project using LSTM and GRU models has provided valuable insights into the capabilities of deep learning models for natural language processing tasks. Both models are capable of accurately predicting the next word in a sentence. However, the results vary based on the dataset used and the specific hyper parameters chosen for each model

The LSTM model, with its ability to remember long-term dependencies, performed well on datasets with longer sentences. On the other hand, the GRU model, with its simplified structure, was found to be computationally more efficient and performed well on datasets with shorter sentences. Both models have their own strengths and weaknesses and choosing the right model for the task at hand ultimately depends on the specific requirements of the problem.

The results have shown that both LSTM and GRU models can produce accurate predictions, with LSTM models having a slight edge in terms of performance on longer sentences. Overall, the results demonstrate the effectiveness of deep learning models in solving complex problems in natural language processing tasks and the potential for further improvement in this field

1. **Limitations**

Concepts that cannot be modified within your project are the limitations

List all the limitations that are applicable to your project with a short description for each

* Limitation 1 : Restricted to dataset

Description: The LSTM and GRU models in my project have been trained on three text corpus datasets. Any sentence given as input which does not exist in the dataset can affect the accuracy of my model. The model would throw random words as predictions which would result in meaningless sentences.

* Limitation 2 : Data Quality

Description: The quality of the input data plays a crucial role in the performance of the models. Out-of-vocabulary words, misspelled words, and non-standard text formats can affect the accuracy of the models.

* Limitation 3 : Hyper parameter Tuning

Description: Finding the optimal hyper parameters for each model can be time-consuming and computationally expensive. Improper hyper parameter tuning can result in overfitting or underfitting of the models.

* Limitation 4 : Computational Resources

Description: Both LSTM and GRU models can be computationally expensive and require significant processing power and memory to train.

* Limitation 5 : Sensitivity to Context

Description: Next word prediction models are highly dependent on the context of the words in a sentence. Models can struggle to accurately predict the next word in sentences that contain multiple possible continuation options.

* Limitation 6 : Bias

Description: The models may contain biases inherited from the training data, such as gender, race, or political biases. It is important to carefully consider the source and quality of the training data to minimize these biases.

1. **Future Scope of the Project**

Write in the form of paragraphs for the following questions.

What is the future of your project?

The future of next word prediction projects is bright and holds significant potential for growth and advancement. With the increasing demand for natural language processing in various industries and applications, next word prediction is likely to play a crucial role in providing users with more personalized and intuitive experiences.

* Incorporating additional data sources: Incorporating additional data sources, such as contextual information or user-specific data, can lead to more accurate predictions and a better understanding of the language.
* Model architecture: The architecture of the models, such as LSTMs and GRUs, can be modified and improved to better handle long-term dependencies and incorporate additional contextual information.
* Integration with other NLP tasks: Integrating the next word prediction model with other natural language processing tasks, such as sentiment analysis or machine translation, can result in more advanced and sophisticated systems.
* Real-time predictions: Real-time predictions can be incorporated into interactive systems, such as chatbots or predictive text input systems, to provide users with immediate feedback and support.
* Multi-lingual models: Multi-lingual models can be developed to support next word prediction in multiple languages, leading to more diverse and inclusive applications.
* Improving robustness: Models can be improved to better handle out-of-vocabulary words, misspelled words, and non-standard text formats, leading to more robust predictions.

These are just a few of the many future scopes for next word prediction projects and demonstrate the ongoing potential for growth and advancement in the field of natural language processing.

Overall, the future of next word prediction projects is promising, and we can expect to see continued growth and innovation in the field. The development of more advanced models will bring us closer to achieving a more human-like understanding of language and provide a foundation for a wide range of exciting new applications.

**REFERENCES**

[1] Bharath K, “Next Word Prediction with NLP and Deep Learning”, Towards Data Science,

August 22, 2020

(<https://towardsdatascience.com/next-word-prediction-with-nlp-and-deep-learning48b9fe0a17bf>)

[2] Name of the person who has written the paper of article, “Title of the article”, Name of journal

where the paper was published, Volume of journal, Date of publishing, link(if it is an

online article)