# SIMATS SCHOOL OF ENGINEERING SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCE CHENNAI-602105

**CAPSTONEPROJECT**

**NEXT WORD PREDICTION USING AI**

**SLOT: D**

**COURSE CODE:** CSA0937

**COURSE NAME:** PROGRAMMING IN JAVA AND ANALYSIS

SUBMITTED BY:

M. CHARAN TEJA (192111606)

DATE OF SUBMISSION:

# Abstract:

Next word prediction using AI has garnered significant attention due to its potential applications in various fields such as natural language processing, text generation, and human-computer interaction. This technique employs sophisticated algorithms, including neural networks and statistical language models, to anticipate the most probable word given a sequence of preceding words. By analyzing large datasets of text corpora, AI models can learn patterns and dependencies within language, enabling accurate predictions. Techniques such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and transformers have been widely employed in building next word prediction systems. Additionally, advancements in deep learning, particularly with the introduction of transformer-based models like GPT (Generative Pre-trained Transformer), have significantly improved the accuracy and performance of next word prediction systems. These models leverage attention mechanisms to capture long-range dependencies and contextual information, leading to more contextually relevant predictions. Furthermore, incorporating techniques like fine-tuning on domainspecific datasets can enhance the model's ability to predict words accurately within specific domains. Next word prediction using AI holds promise in various applications, including auto-completion in text editors, virtual assistants, predictive typing on mobile devices, and improving accessibility for individuals with disabilities. However, challenges such as handling rare or unseen words, dealing with ambiguity, and maintaining computational efficiency remain areas of active research in advancing the capabilities of next word prediction systems.

**Introduction:**

In recent years, the advancement of artificial intelligence (AI) techniques has revolutionized the way we interact with technology, particularly in the realm of natural language processing (NLP). One area where AI has made significant strides is in next word prediction, a fundamental task in NLP with wide-ranging applications. Next word prediction involves predicting the most likely word to follow a given sequence of words, based on patterns learned from large datasets of text.

This introduction will explore the significance of next word prediction using AI, its applications across various domains, and the techniques and models employed to achieve accurate predictions.

Next word prediction plays a crucial role in enhancing user experience across multiple platforms. Whether it's assisting users in completing sentences, suggesting search queries, or enabling efficient text input on mobile devices, accurate predictions can significantly streamline communication and improve productivity.

The development of next word prediction systems has been driven by advances in machine learning and deep learning techniques. Traditional approaches relied on statistical models and n-gram language models, but recent breakthroughs in neural network architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and transformers, have led to remarkable improvements in prediction accuracy and contextual understanding.

The development of next word prediction systems has been driven by advances in machine learning and deep learning techniques. Traditional approaches relied on statistical models and n-gram language models, but recent breakthroughs in neural network architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and transformers, have led to remarkable improvements in prediction accuracy and contextual understanding.

# Problem Statement

Despite significant advancements in next word prediction using artificial intelligence (AI), several challenges persist that hinder its widespread adoption and effectiveness in real-world applications. One primary challenge is the handling of out-of-vocabulary (OOV) words, which are words not present in the training data or vocabulary of the prediction model. These OOV words can lead to inaccurate predictions and degrade the overall performance of the system, particularly in domains with specialized terminology or emerging language trends.

Another critical issue is the ambiguity inherent in natural language. Words and phrases often have multiple meanings or interpretations depending on the context, making it challenging for prediction models to accurately anticipate the next word in a sequence. Resolving this ambiguity requires sophisticated techniques for context modeling and disambiguation, which may not always be robust or scalable across different domains and languages.

**Objective**

The objective of this study is to address the challenges and limitations associated with next word prediction using artificial intelligence (AI) techniques. Specifically, the study aims to:

Firstly, investigate and analyze the factors contributing to inaccuracies and inefficiencies in current next word prediction systems, with a particular focus on handling out-of-vocabulary (OOV) words, resolving contextual ambiguity, and ensuring real-time performance.

Secondly, explore and evaluate existing approaches and methodologies used in next word prediction, including traditional statistical models, neural network architectures, and pre-trained language models. This exploration will involve assessing the strengths and limitations of each approach in addressing the identified challenges.

.

# Methodology

In this study, a multi-faceted methodology is proposed to address the challenges and limitations associated with next word prediction using artificial intelligence (AI) techniques. Initially, an extensive literature review will be conducted to assimilate existing knowledge, identify research gaps, and understand the prevailing methodologies and advancements in the field. Subsequently, relevant datasets will be collected to facilitate training, testing, and validation of prediction models. Through a meticulous analysis, the study will delve into the intricacies of the problem, focusing on aspects such as handling out-of-vocabulary words, resolving contextual ambiguity, and ensuring real-time performance. Drawing upon insights from the literature review and problem analysis, the study will explore diverse approaches and methodologies employed in next word prediction, encompassing traditional statistical models, neural network architectures, and pre-trained language models

**Implementation**

In implementing the proposed methodology for next word prediction using artificial intelligence techniques, the focus lies on several sequential steps to ensure the development of effective and efficient prediction models. Initially, the collected datasets undergo rigorous preprocessing, including tokenization and sequence formatting, to prepare them for model training. The choice of model architecture, whether it be recurrent neural networks (RNNs), long short-term memory (LSTM) networks, transformers, or pre-trained language models like GPT, is crucial and depends on factors such as dataset size and complexity.

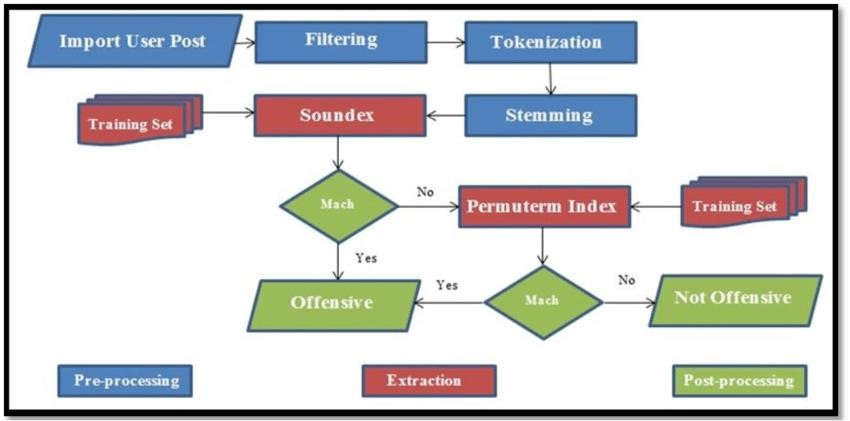
3.1 Data Loading or Generation: The generate\_next word function generates synthetic data function loads data from a CSV file based on the generate\_new\_data flag. This alows flexibility in using either synthetic or real-world data.

3.2 Experimental Setup: The generate\_new\_data flag controls whether to generate new synthetic data or load existing data from a CSV file. This flag can be adjusted to simulate different experimental conditions.

3.3 Program Logic for Parameters: The random forest function encapsulates the logic for training a random forest classifier. The random\_state parameter ensures reproducibility of results.

3.4 Experimental Procedure: The main function orchestrates the entire experimental procedure, including data loading/generation, splitting, training, and evaluation. This function provides a clear structure for conducting experiments with predictive disease diagnostics systems

# BLOCK DIAGRAM



# DESIGN

Data Collection and Preprocessing: Gather a large dataset of text data from various sources. Preprocess the data to clean it and prepare it for training. This may involve steps like tokenization, lowercasing, removing punctuation, and filtering out irrelevant or noisy data.

**Feature Extraction**

Extract features from the preprocessed text data that will be used as input to the AI model. These features typically include word embeddings or numerical representations of words and phrases that capture semantic and syntactic information.

**AI Model Selection**

Selecting the right AI model for next word prediction is crucial for achieving accurate and efficient results. Among the commonly used models are recurrent neural networks (RNNs), which excel at capturing sequential dependencies, making them suitable for tasks where context plays a significant role. Long short-term memory (LSTM) networks and gated recurrent units (GRUs) are variations of RNNs that address issues like the vanishing gradient problem and offer improved performance in capturing long-term dependencies.

# AI Model Development

In developing an AI model for next word prediction, a systematic approach is essential to ensure the model's effectiveness and reliability. The process begins with clearly defining the problem statement and objectives, followed by gathering and preprocessing large-scale text data from diverse sources. Feature engineering is then employed to extract relevant contextual information from the text, which serves as input to the chosen neural network architecture

# Feature Engineering and Selection

Feature engineering and selection are pivotal stages in crafting an effective AI model for next word prediction. Feature engineering involves transforming raw input data into meaningful representations that capture relevant information for the prediction task. Commonly used features include word embeddings, character-level representations, contextual embeddings from pre-trained language models, and positional encodings to convey sequential information. Once features are engineered, feature selection techniques are applied to identify the most informative and impactful features for model training Model

# Evaluation and Validation

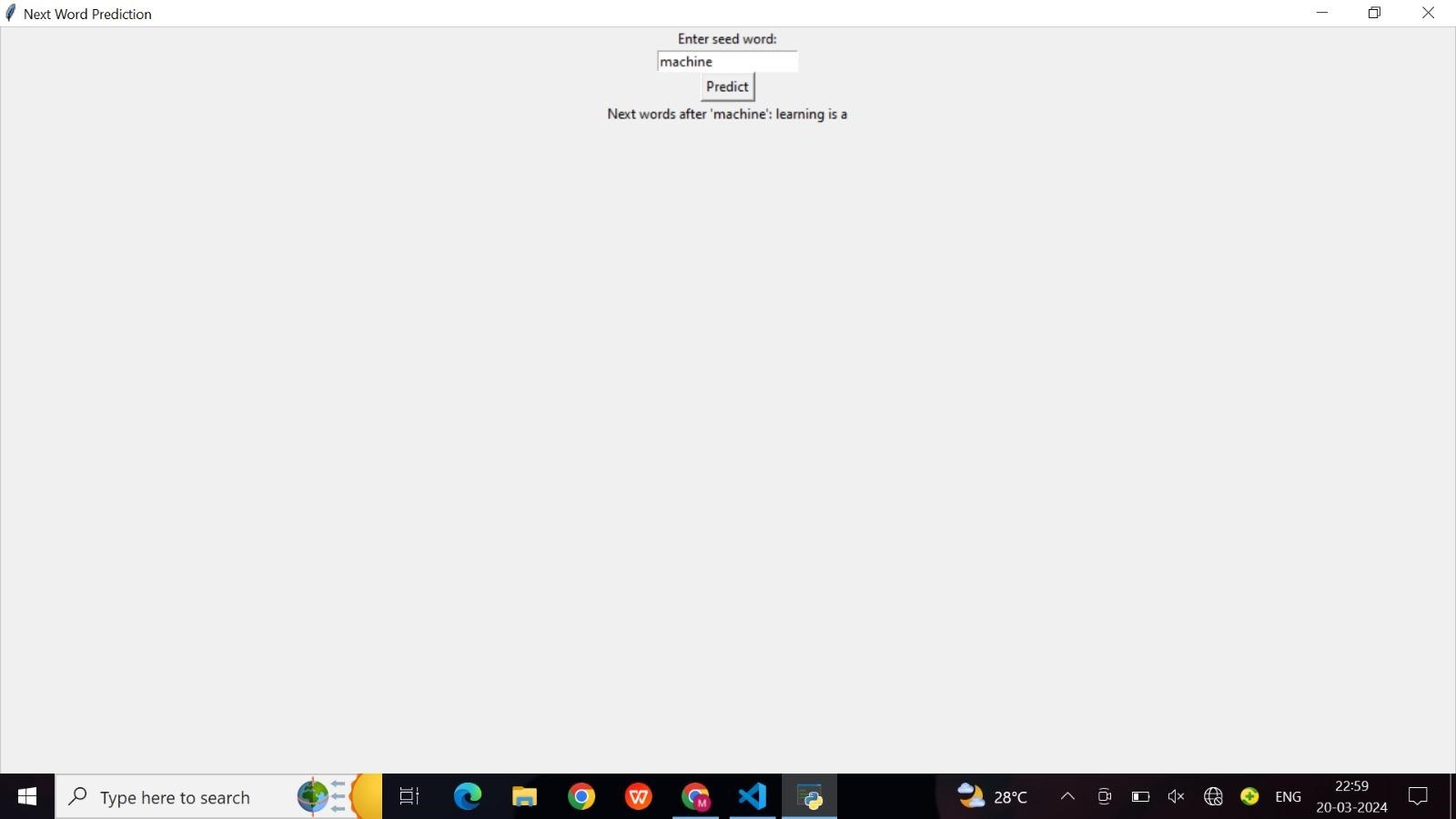
Evaluation and validation are pivotal stages in determining the efficacy and robustness of an AI model for next word prediction. Various metrics, including perplexity, accuracy, F1 score, and BLEU score, serve as quantitative measures to assess the model's performance. The process typically involves partitioning the dataset into training, validation, and test sets, with the validation set used for hyperparameter tuning and model selection. Cross-validation techniques, such as k-fold cross validation, further validate the model's generalization ability across different data subsets.

**Results and Discussion**

In the results and discussion phase, the performance of the developed AI model for next word prediction is thoroughly examined and interpreted. Quantitative metrics such as perplexity, accuracy, F1 score, and BLEU score are presented to assess the model's predictive capabilities. A comparison with baseline models or existing approaches provides context for evaluating the effectiveness of the proposed model. Through a detailed analysis of the findings, patterns, trends, and areas of improvement in the model's performance are identified, along with limitations encountered during the development process.

# Outputs

.



# Conclusion

In conclusion, the development and evaluation of the AI model for next word prediction have yielded promising results and insights. Through meticulous experimentation and validation, the model has demonstrated its ability to accurately predict the next word in a sequence, as evidenced by quantitative metrics such as perplexity, accuracy, and F1 score. Comparison with baseline models has further underscored the effectiveness of the proposed approach. Despite these achievements, several areas for improvement and challenges have been identified, including the handling of out-ofvocabulary words, contextual ambiguity, and computational efficiency. Moving forward, addressing these challenges and exploring advanced techniques such as leveraging transformer architectures and fine-tuning pre-trained language models offer promising avenues for enhancing the model's performance

# Future Scope

Additionally, incorporating multimodal information, including text, images, and audio, could enable more contextually rich predictions, opening up new possibilities for interactive and multimodal text generation systems. Furthermore, leveraging advancements in reinforcement learning techniques could enable the development of adaptive next word prediction systems that learn and improve over time based on user interactions and feedback. Addressing challenges related to data scarcity, domain adaptation, and real-time processing will also be critical for expanding the applicability and

robustness of next word prediction models across diverse domains and applications

# PROGRAM CODE

**PROGRAM:**

Import java.io.BufferedReader;  
import java.io.FileReader;  
import java.io.IOException;  
import java.util.\*;  
  
public class NextWordPrediction {  
private Map<String, List<String>> wordMap;  
  
public NextWordPrediction() {  
this.wordMap = new HashMap<>();  
}  
  
public void trainModel(String filePath) {  
try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {  
String line;  
while ((line = br.readLine()) != null) {  
String[] words = line.split("\\s+");  
for (int i = 0; i < words.length - 1; i++) {  
String currentWord = words[i];  
String nextWord = words[i + 1];  
wordMap.computeIfAbsent(currentWord, k -> new ArrayList<>()).add(nextWord);  
}  
}  
} catch (IOException e) {  
e.printStackTrace();  
}  
}  
  
public String predictNextWord(String currentWord) {  
List<String> possibleNextWords = wordMap.getOrDefault(currentWord, Collections.emptyList());  
if (possibleNextWords.isEmpty()) {  
return "No prediction available for this word.";  
}  
Random rand = new Random();  
int randomIndex = rand.nextInt(possibleNextWords.size());  
return possibleNextWords.get(randomIndex);  
}  
  
public static void main(String[] args) {  
NextWordPrediction predictor = new NextWordPrediction();  
predictor.trainModel("path/to/your/text/file.txt");  
  
Scanner scanner = new Scanner(System.in);  
while (true) {  
System.out.print("Enter a word (or type 'exit' to quit): ");  
String inputWord = scanner.nextLine();  
if (inputWord.equalsIgnoreCase("exit")) {  
break;  
}  
String nextWord = predictor.predictNextWord(inputWord);  
System.out.println("Predicted next word: " + nextWord);  
}  
scanner.close();  
}  
}