Journal of Science, Computing and Engineering Research (JSCER)
Volume-9 , Issue-4 April 2026.
DOI: https://doi.org/10.46379/jscer.2026.060503

SMART ENGLISH TEXT NEXT WORD PREDICTOR

[1]Dr. Anusuya, [2]Mayank Saini, [3]Chetna Soni, [4]Love Kumar Saini, [5]Kush Kumar Saini

[1] Professor, Department of AI&DS, Modern Institute of Technology and Research Centre, Rajasthan, India.

[2,3,4,5] UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Rajasthan, India

Article Information

Received : 29 March 2026

Revised : 30 March 2026

Accepted : 31 March 2026

Published : 01 April 2026

Corresponding Author:

Mayank Saini

Abstract— This project develops a next-word prediction application for English typing using natural language processing techniques. The model is trained on a Wikipedia dataset of approximately 150,000 words to support diverse vocabulary across domains. Preprocessing steps include tokenization, stop word removal, and encoding, while feature extraction uses TF-IDF and tokenization. An LSTM-based model is implemented to capture context and predict the next word accurately. The system can also perform basic arithmetic operations by detecting numerical input and switching functionality accordingly. The trained model is deployed in a mobile application using React and TensorFlow Lite for real-time performance and improved user typing experience.

Keywords: application; LSTM; machine learning; natural language processing; prediction; preprocessing; react; tokenization

Citation: Dr. Anusuya, Mayank Saini, Chetna Soni, Love Kumar Saini, Kush Kumar Saini, "SMART ENGLISH TEXT NEXT WORD PREDICTOR", Journal of Science, Computing and Engineering Research, 9(4), April2026.

I. INTRODUCTION

This project focuses on developing an efficient next-word prediction system that enhances user typing experience through intelligent language modeling. The system is designed to predict the most probable next word based on previously entered text, similar to features found in modern messaging applications.

The approach begins with data analysis and preprocessing, including cleaning, tokenization, and structuring of textual data. These steps ensure that the input is suitable for model training and improves overall prediction accuracy. The project leverages natural language processing techniques along with deep learning methods to build a reliable prediction model.

Two primary approaches are considered for next-word prediction: N-gram models and Long Short-Term Memory (LSTM) networks. N-gram models rely on probabilistic methods such as Markov Chains, where the next word is predicted based on the previous sequence of words. While simple and effective for smaller contexts, this method has limitations in capturing long-term dependencies.

To address this, the project utilizes an LSTM-based neural network, which is more advanced and capable of learning long-range dependencies in text. LSTM models use memory cells to retain important contextual information, making them highly

The system demonstrates how neural networks can significantly improve text prediction by understanding context and sequence patterns. It is designed to be integrated into applications such as mobile keyboards or messaging platforms, providing features like auto-complete and suggested responses.

Overall, the goal of this project is to create a fast, accurate, and adaptive next-word prediction system that improves communication efficiency and showcases the practical use of deep learning and NLP in everyday applications.

II. PROBLEM STATEMENT

Existing mobile keyboard applications provide minimal contextual input and limited word correction capabilities, as they rely mainly on static dictionaries instead of understanding the context of a sentence. Due to this limitation, users often experience slow typing speed, frequent spelling mistakes, and irrelevant word suggestions that do not adapt to their typing behavior. Moreover, most keyboards are not designed for specialized typing scenarios such as programming or technical writing, where predictions must align with syntax and keywords. To address these issues, the objective of this project is to develop an intelligent next-word prediction system that reduces user effort and improves typing efficiency predictive systems, users can now generate text more quickly and easily. Another important objective is to use Natural Language Processing techniques to convert words into vector representations so that the system can understand the meaning and context of words and provide more accurate predictions.

Therefore, the proposed Smart Keyboard will use machine learning-based text prediction implemented through TensorFlow Lite in a Flutter environment to deliver efficient, real-time, and personalized typing experiences.

III. PROPOSED METHOD

The proposed model of the Mental Health Companion Chatbot focuses on developing a smart, secure, and empathetic system that provides timely mental health support. The system integrates Artificial Intelligence (AI), Natural Language Processing (NLP), and Machine Learning (ML) to analyze user input, detect emotions, and generate appropriate responses.

A. Data Processing and Input Handling

The workflow begins when the user sends a message to the chatbot. The system performs text preprocessing, which includes tokenization, cleaning, stop-word removal, and feature extraction. This step ensures that the input text is converted into a structured format suitable for analysis.

B. Emotion Classification

After preprocessing, the cleaned data is passed to a trained Machine Learning model that classifies the emotion of the text. The system categorizes emotions into classes such as stress, happiness, sadness, anxiety, anger, or neutrality. This helps in understanding the mental state of the user.

C. Risk Detection Mechanism

A risk detection module analyzes whether the user's message indicates severe distress, self-harm, or suicidal thoughts. Based on the level of risk detected, the system decides the type of response to generate. This ensures user safety and timely intervention in critical situations. The risk detection algorithm assigns a risk level.

D. Response Generation

For normal emotional conditions, the chatbot generates empathetic and supportive responses based on psychological principles like active listening and reassurance. In high-risk situations, the system activates a crisis response mode, providing coping techniques, breathing exercises, grounding methods, and helpline support.

E. Mood Tracking Module

The system includes a mood tracking feature where users can log their emotions manually or automatically through chat analysis. These records are stored in a database and help in tracking emotional patterns over time. Users can also view their mood history for better self-awareness.

F. Assessment Module

The chatbot includes mental health assessment tools such as PHQ-9 for depression and GAD-7 for anxiety. These assessments help evaluate the severity of the user's condition and categorize it into different levels, enabling better recommendations.

G. Recommendation System

A recommendation engine analyzes user data, mood history, and assessment results to provide personalized suggestions. These may include meditation, journaling, physical exercise, or lifestyle improvements based on the user's needs

.

H. Data Storage and Security

All user interactions, mood logs, and assessment data are securely stored in the database. Proper security measures are implemented to ensure data privacy and confidentiality, making the system reliable and safe for users.

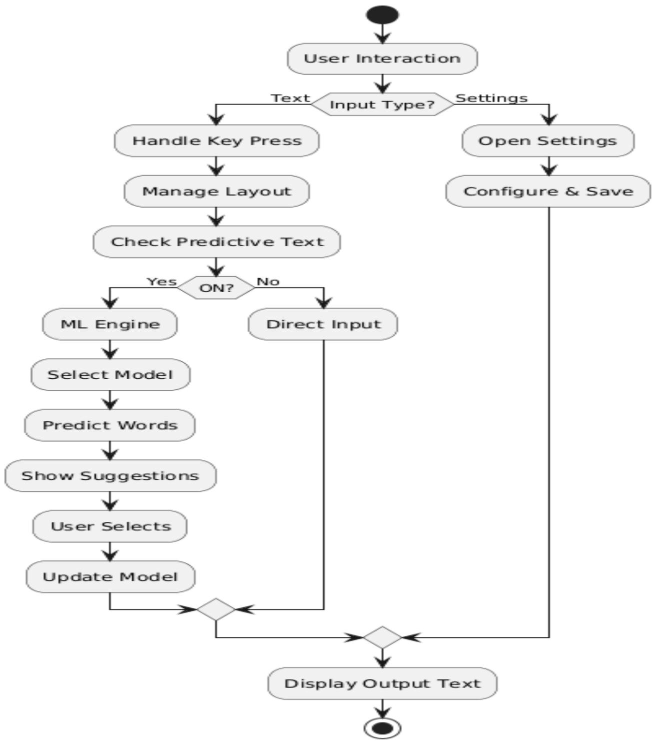

Fig. 1. Proposed Work Model

IV. TECH STACK

A. Frontend Technologies

The frontend of the system is developed using HTML5, CSS3, JavaScript, and Bootstrap to create a responsive, interactive, and user-friendly interface. HTML5 is used to structure web pages and define content with improved semantic elements and multimedia support. CSS3 is used for styling and layout design, providing features such as animations.

B. Backend Technologies

The backend of the system is built using Django, a high-level Python web framework that supports rapid development and clean design. Django handles core functionalities such as user request processing, URL routing, database interactions, and authentication, managing data flow and system operations efficiently.

C. Artificial Intelligence and Machine Learning

The intelligent capabilities of the system are powered by TensorFlow and Keras, which are used to build and train the sentiment analysis model based on deep learning techniques such as LSTM (Long Short-Term Memory). TensorFlow provides a comprehensive framework for model development, training, and deployment, while Keras simplifies the process with a high-level API for designing neural networks.

D. Natural Language Processing Tools

To handle multilingual input and improve text understanding, the system uses libraries such as Googletrans and Langdetect. Googletrans is used for translating user input into a common language, enabling consistent processing of text data. Langdetect is used to identify the language of the input text before translation. These tools enhance the system's ability to process diverse inputs and ensure accurate sentiment analysis across different languages.

E. Data Storage Technologies

The system uses SQLite as the primary database for storing application data, including user inputs, results, and system logs. SQLite is a lightweight, serverless database that is easy to configure and suitable for small to medium-scale applications. In addition, Pickle is used for serializing and deserializing Python objects such as trained models, tokenizers, and preprocessed datasets.

F. Machine Learning Preprocessing Tools

Scikit-learn is used for data preprocessing tasks such as label encoding, data splitting, and feature preparation. It provides efficient tools for preparing datasets before training machine learning models. Preprocessing ensures that the input data is clean, well-structured, and suitable for model training.

G. Deployment Technologies

For deployment, the system uses WSGI (Web Server Gateway Interface) and ASGI (Asynchronous Server Gateway Interface). WSGI is used for handling synchronous web requests and acts as a bridge between the web server and the Django application. ASGI extends this capability by supporting asynchronous operations.

H. Additional Tools and Utilities

The system also uses additional tools such as PyCountry for managing country and language-related information. PyCountry provides standardized data for languages and regions, which helps in improving localization and user interaction. These supporting tools enhance the overall functionality, usability.

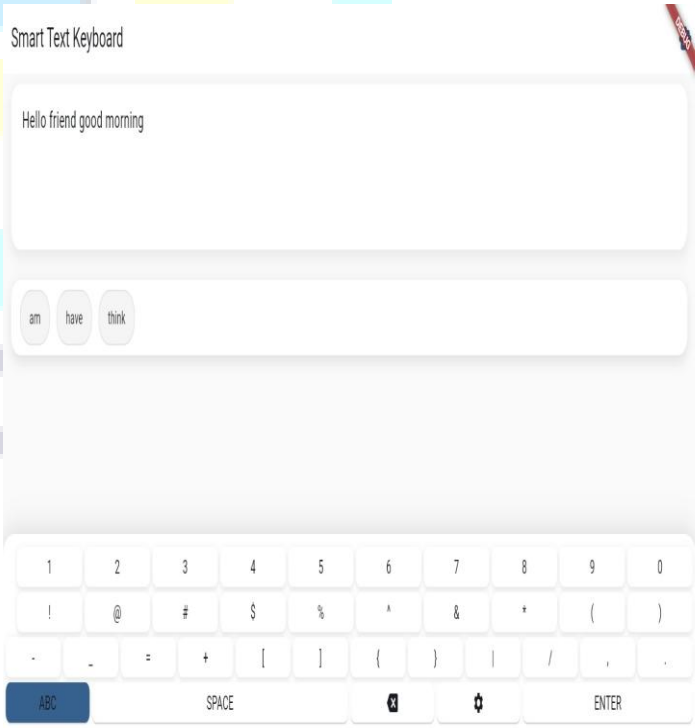

Fig. 2. Next Word Predictor

V. RESULTS

The developed Next Word Prediction system successfully demonstrates the practical application of Artificial Intelligence and Machine Learning in enhancing typing efficiency. The system is capable of providing real-time, context-aware word predictions using an LSTM model optimized with TensorFlow Lite for mobile deployment. It generates accurate next-word predictions as well as top-K suggestions, improving user typing speed and reducing effort. Integration with a React Native mobile application enables smooth and responsive interaction, while offline prediction capability ensures faster performance and better data privacy. The model effectively captures sequential dependencies in text and performs well for common language inputs, although it may face limitations with rare words, complex sentences, and domain-specific vocabulary. Overall, the results show that the proposed system functions efficiently as an intelligent text prediction tool, offering a scalable solution that can be extended to multilingual and personalized typing experiences.

VI. CONCLUSION

The proposed Text Flow Keyboard system demonstrates how a machine learning–based predictive typing application can provide a more adaptive, efficient, and intelligent user experience compared to traditional keyboard systems. By integrating LSTM-based next word prediction, contextual understanding, swipe and tap input processing, and real-time analysis within a Flutter framework, the system delivers fast and accurate text suggestions while maintaining a smooth and user-friendly interface. The use of on-device processing and privacy-focused techniques ensures that user data remains secure while still allowing continuous model improvement. The workflow enables dynamic prediction updates based on user input, improving typing speed and reducing effort. Additionally, features such as customizable settings, scalable architecture, and responsive state management make the system flexible and maintainable. Overall, the model provides a practical and scalable solution for intelligent text input, which can be further extended with multilingual support, advanced prediction techniques, and deeper personalization, highlighting the potential of AI-driven keyboard systems to enhance communication efficiency and user productivity.

REFERENCES

[1] B. Inkster, S. Sarda, and V. Subramanian, “An empathy-driven conversational artificial intelligence agent (Wysa) for digital mental well-being: Real-world data evaluation,” *JMIR mHealth and uHealth*, vol. 6, no. 11, p. e12106, 2018.

[2] R. Fulmer, A. Joerin, B. Gentile, L. Lakerink, and M. Rauws, “Using psychological artificial intelligence to support mental health,” *Frontiers in Psychology*, vol. 9, p. 797, 2018.

[3] A. Abd-alrazaq, M. Alajlani, A. Alalwan, B. Bewick, P. Gardner, and M. Househ, “An overview of the features of chatbots in mental health: A scoping review,” *International Journal of Medical Informatics*, vol. 150, pp. 104–129, 2021.

[4] A. N. Vaidyam, H. Wisniewski, J. D. Halamka, M. S. Kashavan, and J. B. Torous, “Chatbots and conversational agents in mental health: A review of the psychiatric landscape,” *The Canadian Journal of Psychiatry*, vol. 64, no. 7, pp. 456–464, 2019.

[5] World Health Organization, *Digital tools for mental health improvement*. WHO Publications, 2022.

[6] A. Ho, J. Hancock, and A. S. Miner, “Psychological effects of AI tools in mental health,” *Journal of Medical Internet Research*, vol. 20, no. 11, p. e11293, 2018.

[7] OpenAI, “Using AI models for mental health conversations: Ethical and technical considerations,” *OpenAI Research Notes*, 2023.

[8] C. A. Figueroa *et al.*, “AI-based conversational agents in mental health: Current practices and future trends,” *Nature Digital Medicine*, vol. 4, no. 1, pp. 1–12, 2021.

[9] HealthIT.gov, “Privacy and security considerations for digital mental health applications,” U.S. Department of Health & Human Services, 2022.

[10] P. Kumar and D. Ghosh, “AI-based mental health chatbots: A review of technologies, models, and applications,” *International Journal of Artificial Intelligence Research*, vol. 15, no. 2, pp. 55–70, 2021.