

Uni-NLP: An Integrated Framework for Multi- Domain Natural Language Processing

Mr. P. Ilanna¹, Juluru Chaitanya Sai², Tanaji Anudeep³, Agnal Napoleon⁴, K. Sai Koushik⁵

Assistant Professor, Department of CSE (Data Science), ACE Engineering College, Ghatkesar, Telangana, India¹

U.G. Student, Department of CSE (Data Science), ACE Engineering College, Ghatkesar, Telangana, India²

U.G. Student, Department of CSE (Data Science), ACE Engineering College, Ghatkesar, Telangana, India³

U.G. Student, Department of CSE (Data Science), ACE Engineering College, Ghatkesar, Telangana, India⁴

U.G. Student, Department of CSE (Data Science), ACE Engineering College, Ghatkesar, Telangana, India⁵

ABSTRACT: The rapid advancement of digital technologies and artificial intelligence has significantly improved the way humans interact with computers through natural language processing systems. This project presents **UNI-NLP: An Integrated Framework for Multi-Domain Natural Language Processing**, a web-based platform designed to perform multiple language processing tasks within a unified environment. The system enables functionalities such as text summarization, translation, and intelligent text generation through a user-friendly interface. The proposed system integrates modern web technologies including **React.js and TypeScript for the frontend, Node.js and Express.js for the backend**, to perform advanced NLP operations. In addition, the system uses **PostgreSQL with Drizzle ORM** for efficient data storage and session management. By combining multiple NLP capabilities into a single platform, the system reduces dependency on separate tools and improves workflow efficiency.

KEYWORDS: Natural Language Processing, React.js, Node.js, Python libraries.

I. INTRODUCTION

Natural Language Processing (NLP) has become an important area of artificial intelligence that enables computers to understand, interpret, and generate human language effectively. With the rapid growth of digital technologies and online communication platforms, the demand for intelligent language processing systems has increased significantly. NLP applications are widely used in areas such as text summarization, translation, chatbots, content generation, and automated data analysis. These systems help users process large volumes of textual information quickly and efficiently. However, many existing NLP tools focus on only a single functionality, requiring users to switch between multiple platforms to perform different language processing tasks. To overcome these limitations, this project introduces **UNI-NLP: An Integrated Framework for Multi-Domain Natural Language Processing**, a unified web-based platform designed to perform multiple NLP operations within a single system.

The proposed system integrates modern web technologies and intelligent language processing techniques to provide functionalities such as text summarization, translation, and advanced text processing through a user-friendly interface. The application is developed using **React.js with TypeScript and Tailwind CSS for the frontend**, while the **backend is implemented using Node.js and Express.js** to manage API communication and application logic. The system also uses **PostgreSQL with Drizzle ORM** for efficient data storage and session management. By combining multiple NLP features.

II. LITERATURE SURVEY

Recent research explores the development of intelligent Natural Language Processing (NLP) systems that integrate multiple language processing capabilities such as translation, summarization, and text generation. Transformer-based architectures have significantly improved the performance of modern NLP applications by enabling contextual understanding of language across different domains. [1] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proc. ACL, 2020, pp. 7871–7880. [2] C.

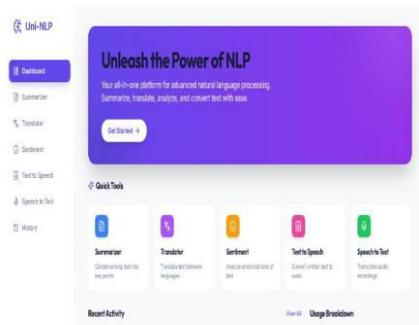
Raffel et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020. [3] A. Conneau et al., “Unsupervised cross-lingual representation learning for multilingual language understanding,” in *Proc. NeurIPS*, 2020 [4] A. Radford et al., “Robust speech recognition via large-scale weak supervision,” OpenAI, 2023 [5] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. NAACL*, 2019, pp. 4171–4186. [6] Y. Liu et al., “RoBERTa: A robustly optimized BERT pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019 [7] T. Wolf et al., “Transformers: State-of-the-art natural language processing,” in *Proc. EMNLP*, 2020, pp. 38–45. [8] J. Brown et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020. [9] T. Vaswani et al., “Attention is all you need,” in *Proc. NeurIPS*, 2017, pp. 5998–6008. The literature highlights several important challenges in modern NLP systems. Many applications still rely on separate tools for individual language tasks, which reduces efficiency and workflow integration. Additionally, issues related to scalability, model bias, and computational resource requirements remain significant concerns. Researchers emphasize the need for unified platforms that combine multiple NLP functionalities within a single framework while maintaining accuracy, usability, and system performance.

III. METHODOLOGY

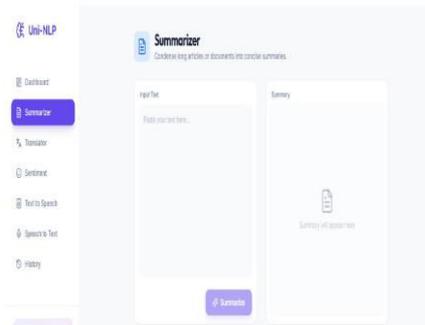
The methodology describes the complete workflow followed to design, develop, and implement the proposed UNI-NLP system. It includes a structured step-by-step process beginning from requirement analysis and system design to implementation, integration, testing, and deployment. The development of the proposed NLP platform follows a systematic methodology to ensure efficiency, scalability, reliability, and ease of use. The process begins with requirement analysis, where both functional and non-functional requirements of the system are identified and documented. In this stage, existing Natural Language Processing tools, web frameworks, and intelligent language processing techniques are studied to understand their capabilities and limitations. User interactions such as text input, request submission, and output visualization are analyzed to determine how users will interact with the system. The methodology describes the complete workflow followed to design, develop, and implement the proposed UNI-NLP system. It includes a structured step-by-step process beginning from requirement analysis and system design to implementation, integration, testing, and deployment. The development of the proposed NLP platform follows a systematic methodology to ensure efficiency, scalability, reliability, and ease of use. The process begins with requirement analysis, where both functional and non-functional requirements of the system are identified and documented. In this stage, existing Natural Language Processing tools, web frameworks, and intelligent language processing techniques are studied to understand their capabilities and limitations. User interactions such as text input, request submission, and output visualization are analyzed to determine how users will interact with the system. The backend server manages API communication, request processing, and system logic while interacting with the database to store application data and user sessions. The next phase involves the implementation of the system using modern web development technologies. The frontend interface is developed to provide a responsive and user-friendly environment where users can enter text, request language processing tasks, and view the results generated by the system. The backend server manages API communication, request handling, and system logic, ensuring efficient processing of user requests. The backend also interacts with the database to store and retrieve application data, user sessions, and processing history. Data validation and error handling mechanisms are incorporated to ensure system reliability and accuracy. In addition to system development, testing and performance evaluation are conducted to verify the functionality of the application. Different test cases are executed to ensure that the system processes user inputs correctly and generates accurate outputs. System integration testing is performed to ensure smooth interaction between frontend, backend, and database components. Finally, the application is deployed in a web environment where users can access the platform and perform multiple language processing tasks within a unified system. This methodology demonstrates a practical implementation of Natural Language Processing technologies integrated with modern web development.

IV. EXPERIMENTAL RESULTS

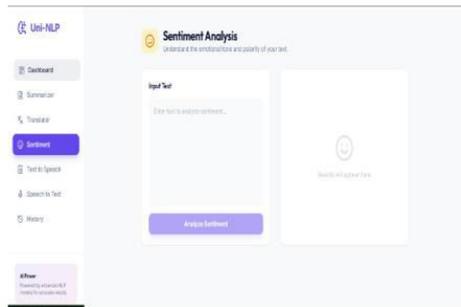
The below images showcase the output screens that have been obtained from the code implementation and thus rendering the desired results.



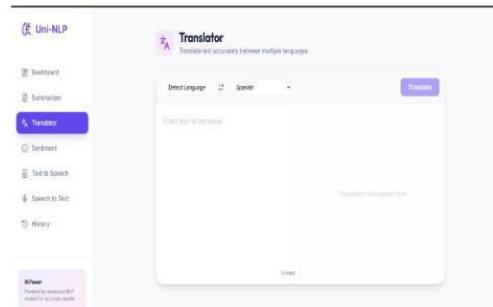
(a)



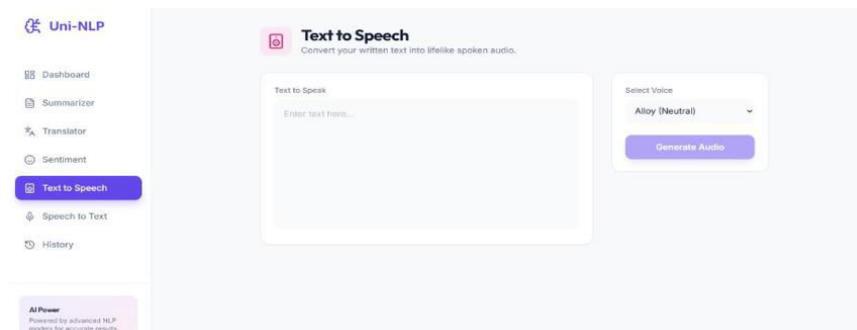
(b)



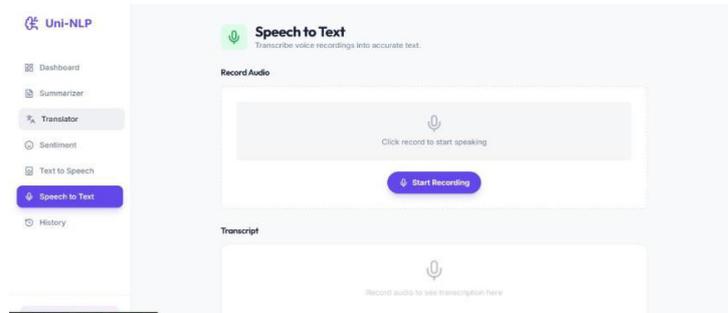
(c)



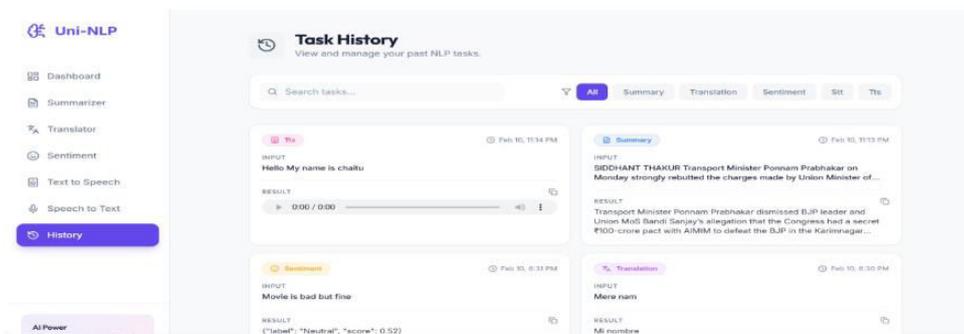
(d)



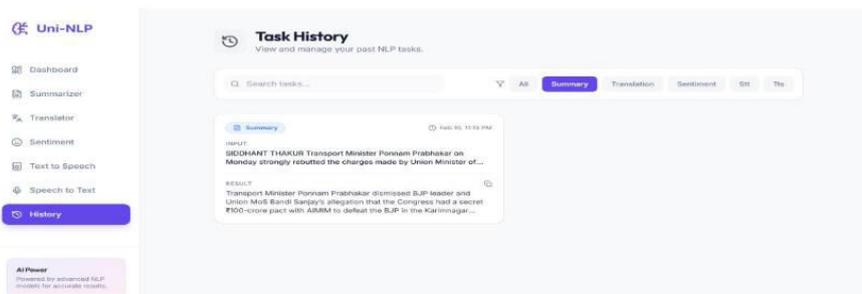
(e)



(f)



(g)



(h)

The first screen (a) shows the **main dashboard interface of the UNI-NLP platform**, which acts as the central control panel for accessing different Natural Language Processing tools. The page displays the heading **“Unleash the Power of NLP”** along with a short description explaining that the system provides an all-in-one platform for advanced language processing tasks. The dashboard contains a sidebar navigation menu with different modules including **Dashboard, Summarizer, Translator, Sentiment Analysis, Text to Speech, Speech to Text, and History**. Each module represents a specific NLP functionality available within the system. The dashboard also includes **Quick Tools** cards that provide shortcuts to commonly used features such as summarization, translation, sentiment analysis, and speech processing. This layout helps users easily navigate through the platform and access the required functionality without complexity.

The next screen (b) displays the **Summarizer module**, which is designed to convert lengthy text documents into shorter and meaningful summaries. The interface is divided into two main sections: the **Input Text panel** and the **Summary output panel**. In the input panel, users can paste or type long articles, reports, or paragraphs that need to be summarized. Once the user clicks the **“Summarize”** button, the system processes the text using natural language

processing techniques to identify key sentences and essential information. The generated summary is then displayed in the output section on the right side of the screen. This module helps users quickly understand large documents without reading the entire content.

Another screen (c) illustrates the **Sentiment Analysis module**, which is used to analyze the emotional tone or polarity of a given text. Users can enter a sentence, review, or paragraph into the input text area and click the “**Analyze Sentiment**” button. The system evaluates the input text and determines whether the sentiment expressed is **positive, negative, or neutral**. The results are then displayed in the output panel along with a confidence score indicating the strength of the sentiment prediction. This feature can be useful for analyzing customer feedback, social media posts, or product reviews.

The following screen (d) shows the **Translator module**, which allows users to translate text between multiple languages. The interface provides options for **detecting the source language automatically** and selecting the **target language** from a dropdown menu. Users can enter text in the input area, choose the desired language, and click the “**Translate**” button to perform the translation process. After processing, the translated text is displayed in the output panel. This module is useful for communication across different languages and enables users to understand foreign language content more easily.

The next screen (e) demonstrates the **Text-to-Speech module**, which converts written text into spoken audio. In this interface, users can type or paste text into the input area and choose a voice type from the available options. Once the “**Generate Audio**” button is clicked, the system processes the text and produces a natural-sounding audio output. The generated audio can be played directly within the application. This module is particularly helpful for users who prefer listening to information rather than reading it.

The following screen (f) shows the **Speech-to-Text module**, which converts spoken audio into textual format. The interface provides a recording feature where users can click the “**Start Recording**” button and speak into the microphone. The system captures the voice input and processes it to generate a transcript of the spoken content. The transcribed text appears in the transcript section of the interface. This feature can assist users in converting spoken conversations, lectures, or voice notes into written text.

The next screens (g) and (h) display the **Task History module**, which stores records of previously executed NLP tasks. This interface allows users to view and manage their past operations such as summaries, translations, sentiment analysis results, and speech processing tasks. Each task entry shows the **input text, generated output, and timestamp** indicating when the task was performed. The system also provides filtering options so users can search tasks based on categories such as **Summary, Translation, Sentiment, Speech-to-Text, or Text-to-Speech**. This feature improves usability by allowing users to revisit or reuse previously generated results without repeating the same process.

V. CONCLUSION

The **UNI-NLP system** provides a modern, efficient, and integrated approach for performing multiple Natural Language Processing tasks within a single web-based platform. By combining functionalities such as **text summarization, translation, sentiment analysis, text-to-speech, and speech-to-text**, the system allows users to process and analyze language data more efficiently. The unified interface ensures that users can access different NLP tools from one environment, reducing the need to switch between multiple applications and improving overall workflow productivity.

This system enhances **usability, accessibility, and efficiency** in language processing applications. Users can easily interact with the platform through a simple and responsive interface while the system processes input data and generates meaningful results. Additionally, the integration of modern web technologies and intelligent language processing techniques helps the system deliver faster responses and improved user experience compared to traditional standalone tools.

However, challenges such as **system scalability, processing performance, and accuracy of language analysis** must be carefully considered to maintain reliable results. Proper optimization techniques, efficient data management, and continuous improvements in language models can further enhance the effectiveness and performance of the system.

In conclusion, the proposed **UNI-NLP platform** demonstrates how Natural Language Processing technologies can be effectively integrated with modern web applications to create a powerful, scalable, and user-friendly solution for multi-domain language processing tasks.

REFERENCES

- 1. Ansar, W., et al., “A Survey on Transformers in NLP with Focus on Efficiency.”**
arXiv Research Paper, 2024.
This study analyzes modern transformer-based architectures used in Natural Language Processing and discusses techniques for improving efficiency, scalability, and performance in language processing tasks.
- 2. Wang, C., et al., “A Survey on Large Language Models from Concept to Implementation.”**
arXiv Research Paper, 2024.
This research examines the development of large language models and their applications in text generation, translation, and intelligent conversational systems.
- 3. Wu, T., et al., “Advancements in Natural Language Processing: Exploring Transformer-Based Architectures for Text Understanding.”**
Artificial Intelligence Research Paper, 2025.
This study explores recent advancements in transformer-based NLP models such as BERT and GPT and highlights their effectiveness in language understanding and text processing tasks.
- 4. “A Combined Encoder and Transformer Approach for High-Quality Text Generation.”**
arXiv Research Paper, 2024.
This work proposes a hybrid model combining semantic encoding with transformer-based generation techniques to improve coherence and accuracy in natural language generation.
- 5. “Ravichandran, R., et al., “Text Summarization Using the T5 Transformer Model.”**
International Research Journal of Engineering and Technology (IRJET), Vol. 10, Issue 08, 2023.
This study proposes a text summarization system using the T5 transformer model to extract key information from long documents and generate concise summaries for efficient information understanding.
- 6. Awasthi, S., et al., “Text Summarization using Natural Language Processing.**
International Research Journal of Engineering and Technology, Vol. 11, Issue 03, 2024.
This research explains the use of Natural Language Processing techniques such as TF-IDF and extractive summarization to reduce large textual content into shorter meaningful summaries.
- 7. Reddy, P., et al., “Artificial Intelligence Powered PDF Summarization with NLP.”**
International Research Journal of Engineering and Technology, Vol. 12, Issue 02, 2025.
This study presents an AI-based system that extracts text from PDF documents and applies NLP techniques to summarize and translate the content automatically.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details