JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-POWERED SENTENCE TO STORY GENERATOR

[1]Dontha Manjusha, [2]Y. Sudheer Kumar

[1]Department of Information Technology and Computer Applications, Andhra University, Visakhapatnam

[2] Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam

Abstract : Storytelling has always played an important role that we share knowledge, express imagination, and connect with others emotionally. At a time when technology is developing much faster than ever, I saw the opportunity to bring that traditional art into a more modern, interactive format. The project was born with a simple idea: what if a person can just give a sentence, and a whole story will be made from it - not only in English but also in different languages and styles imagination, mystery, or romance?
This idea created a web-based story generator that does not stop only when the text is generated. The one who stands this tool is the ability to understand the input spoken and also tell the story back to the user in a natural, human voice. It is designed using react.js to keep the interface clean and easy to use, and it uses Google Gemini Pro as the brain behind the generation of the story, ensuring that the output looks meaningful and creative. The web speech API handles voice input and output so that users can speak instead of typing, and listen instead of reading - can also make the app accessible with reading or typing difficulties.

Keywords - Deepseek(R1), story generation, multilingual, speech recognition, React, Web Speech API, text-to-speech

1. INTRODUCTION

In today's digital age, Artificial Intelligence is no longer limited to technical fields - it is also being used in creative domains such as story, music and art. A region that caught my interest was how companies are using AI-based creativity tasks to evaluate candidates. For example, companies such as Infosys have introduced a unique approach during the interview where candidates are given a simple sentence and are asked to think creatively to create a full story around it. This idea fascinated me because it not only tests creativity, but also encourages quick thinking and imagination. Inspired by this, I wanted to create a smart, interactive application that brings this concept into a real -world device. The idea was that users allow either a small sentence to type or speak, choose their style of choice and language, and then let AI make an attractive story with it. What makes this project makes this speech understanding the input and using a natural-sounding voice output back. It combines AI-managed story generations with modern web technologies to tell an accessible, multilingual and voice-competent story. The goal is not only fun and attractive, but also interactive and inclusive for all.

2. PROBLEM ANALYSIS

In recent years, Artificial Intelligence has shown remarkable progress in generating human lessons. However, most of the story generation tools are limited in scope - they usually support English, lack creative flexibility, and do not provide an interactive experience for users. In addition, these devices often depend only on text inputs, which prohibit access to users that may like or require voice interactions. There is also a significant difference in multilingual story systems. Many users around the world want to connect with AI tools in their native languages, but very few platforms support it. In addition, the lack of natural-sounding voice output makes the story stories experience less immersive and attractive. Another important issue is the absence of devices that add to the style selection, language flexibility, voice input and voice output to a spontaneous interface. Users want to control the tone and style of the story, while enjoying the convenience of talking to the app and listening to the story in human -like voice. The purpose of this project is to address these problems: Providing a multilingual, style-specific story production system. To enable voice-based interaction using speech recognition. Adding speech synthesis with natural voice output. Creating a user -friendly web interface that is accessible and responsible...

3. SYSTEM REQUIREMENTS

3.1 **Hardware Requirements**:

System type : 64-bit operating system, x64-based processor, i5

Hard Disk : 512 GB.

processor : AMD Ryzen , 55625U with Radeon Graphics

Input Devices : Keyboard, Mouse

RAM : 16.0 GB

3.2 Software Requirements:

Operating System : Windows 11

Browser : any browser supporting Google Chrome, Microsoft Edge, or Web Speech API

Frontnds : react.js, bootstrap

Backnd : Openrouter api through Google Gemini Pro

Library/Package : Axios - API for requests
React-speech-riknishan-voice input
window.speechSynthesis – for voice output

Node.js : v14 or above

NPM/Yarn : For Package Management

4. METHODOLOGY:

The functioning of this project revolves around the design and development of a multilingual AI-run short story production system using the Deepseek (R1) language model through Openrouter API. The approach is divided into several well-defined stages: input processing, early construction, AI model interaction, output generation, multilingual text-to-speech synthesis and dynamic story extensions. Each stage is carefully orchestrated to ensure spontaneous functionality and natural language coherence in separate languages and styles. This process begins with user interactions on the front -made interface built with react.js, where the user is motivated to enter a sentence that serves as seeds for the story generation. Additionally, the user selects a story style (such as adventure, horror, romance, etc.) and a language (which is not limited to English, Hindi, Telugu, Tamil and French). These inputs form the basis of the prompt that will be sent to the AI model. The system creates a dynamic and descriptive signal in the natural language format, ensuring that it includes sentences, selected style and language instructions, which the model can understand and can effectively respond. Once the prompt is ready, it is sent to the Deepseek (R1) model via post request using Openrouter API. The prompt is carefully wrapped in to direct the model in creating a short story of up to 500 words in the chosen language and style. The Deepseek R1 model processes the input and produces a consistent, relevant rich story. The response is returned as a plain text, which is later parse and formatted in a clean paragraph for better readability. To increase the user experience and access, the application integrates the text-to-speech (TTS) functionality using web speech API.

4.1 Deepseek R1 - Big Language Model

Core algorithm: Transformer-based deep teaching model

Function: User produces consistent, creative stories based on signals.

Use in the project: The sentence provided by a user limits the style and language to 500 words in a meaningful and creative story.

4.2 Speech recognition - web speech API :

Algorithm: On-device speech-toxt using browser-origin model (operated by deepspecheles or similarly hidden deep learning models).

Objective: allows users to input a sentence using their voice instead of typing.

Language support: English, Hindi, Telugu, Tamil, French, German (depending on the browser support).

4.3 Speech Synthesis – window.speechSynthesis API:

Algorithm: Nerve text-to-speech (TTS) using browser-supported synthetic sounds.

Objective: Reads the story generated in the selected language vigorously.

voice control: A woman matches the language selects the voice (if available). If the input changes, it stops reading.

4.4 Early engineering (technology)

Method: You create signs like:

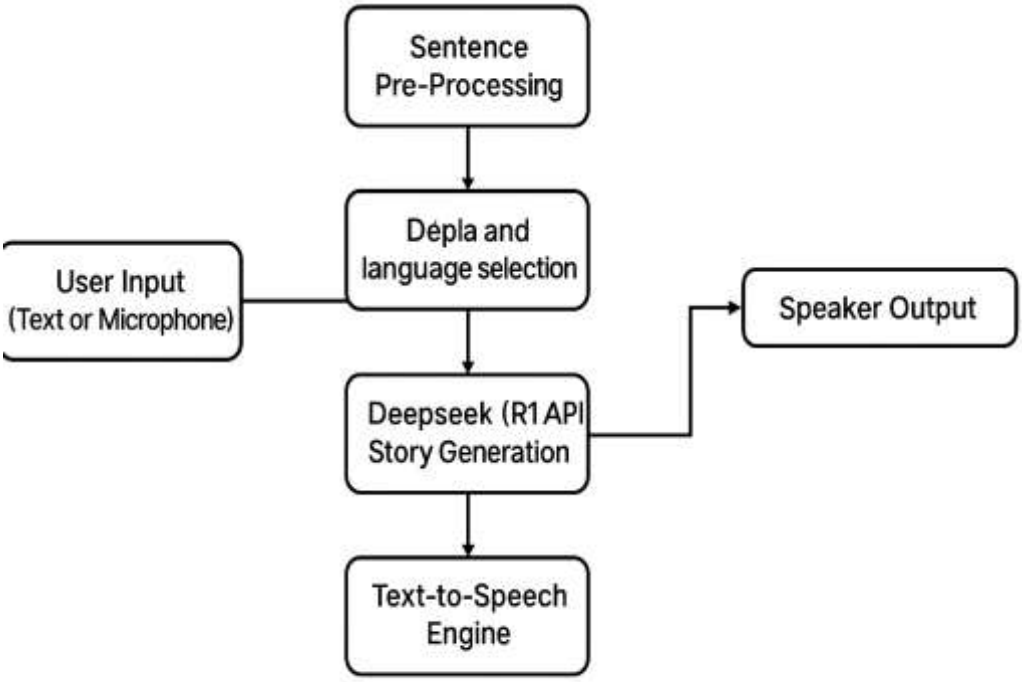
"Depending on the sentence, create a story in [language] in the style of [style]: [user sentence]"

Objective: Guidance of Deepseek models to give proper response to tone, language and materials.

Benefits: No additional model is required for fine-tuning.

5. SYSTEM ARCHITECTURE:

The architecture of the AI-Interacted Sentence-to-story generator is designed to convert a sentence provided by a simple user to a creative and consistent story using advanced natural language processing techniques. The system is made up of an responsible friendly developed using reacts, where users input a sentence, select a story style, and choose a target language. This input is alternatively captured via voice and is then processed in the backend, which forms a structured prompt and communicates with a large language model-as to generate a story of Deepseek R1, GPT-3, or a fine-tund MT5 model-500 words. Once generated, the story is returned to the front for the display and can also be read vigorously using a multilingual text-to-skie engine, which distributes readable and audio story experiences in the selected language.



System architecture

The system architecture consists the following:

5.1 User input (mic or text):
The user interacts with the application by:

- Type a sentence in a text box, or Using a microphone to speak the sentence, which is then converted into a text using a speech-to-read (STT) engine (eg, web speech API or whisper).
- This input story becomes seed idea or initial point for generation.

5.2 Sentence preprocessing:

- In sentence preprocessing the unnecessary data will be remove.
- grammar will be check if necessary.
- This step is very helpful to generator the creative stories

5.3 Style and language selection:
Choose the user:

- A genre or style for the story (eg, horror, adventure, fantasy)

- One language that can generator and spoken.
- This selection is used to modify the prompt sent to the model, guiding it to generate a story in cosen tone and language.

5.4 API request for Deepseek R1 model:
Backnd uses a sign:

- Clean sentence
- Selected style
- Desired language
- It then sends the Deepseek R1 model (or other LLMs like GPT -3) to this prompt through API request, requests a story based on input.

5.5 Display the story on UI:

- Once the story is generated then it goes to the frontend
- It display on the react wed app and reader able text

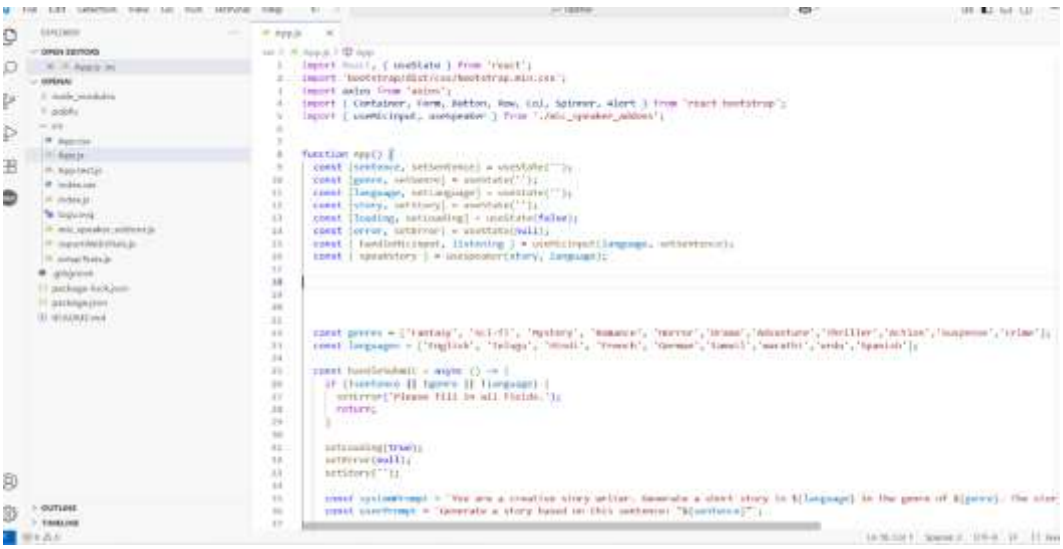
5.6 Use Text-to-Spoken Engine:

- When the user enter the sentence say by using the microphone.
- The Text-to-Speech (TTS) engine is a software that:
- Takes written text as input
- Converts it into a natural-ridden speech
- It plays loudly through the user's device (speaker or headphone).

5.7 Story Playback (Speaker output) in selected language:
Finally, TTS output is played through a browser or system speaker in the user-selection language, allowing the user to hear the story.
This makes the app accessible for users who prefer audio or are visually.

6. IMPLEMENTATION:

The multilingual short story generator is applied as a react-based web application that integrates the Deepseek (R1) large language model through the Openrouter API. The system is designed to generate creative stories based on sentences provided by the user, style selection and language preference. It also includes characteristics for real-time voice statement using web speech API and users are allowed to expand stories dynamically using the "Add content" option. Users interact with Frontend UI to enter a sentence, select a style (such as adventure, horror, or romance), and choose a target language (such as English, Hindi, Telugu, Tamil, or French). When clicked on the "Generate" button, the app creates a well -structured signal from the input and sends it to the Deepseek R1 model via Openrouter API. The Deepseek R1 model processes the prompt and gives a creative, human story, which is pars and display in paragraph format for better readability. The application involves argument for forms of reaction using line and paragraph division. Text-to-speech (TTS) is applied using web speech API, which selects the user's selected language system voice. This enables a clear and natural reading of the output of the story. "Add the content" button the story generates back to the model as a reference and asks it to continue the story, making it more interactive and dynamic. The state hooks of the react are used to manage input values, loading states, generated materials and speech playback status. Frontend is styled using responsible layout techniques to ensure compatibility in equipment. The entire system runs client-side, with all AI interactions, the operator is safely rooted via API & Point. Overall, implementation Shoka



Original code for sentence to story generator

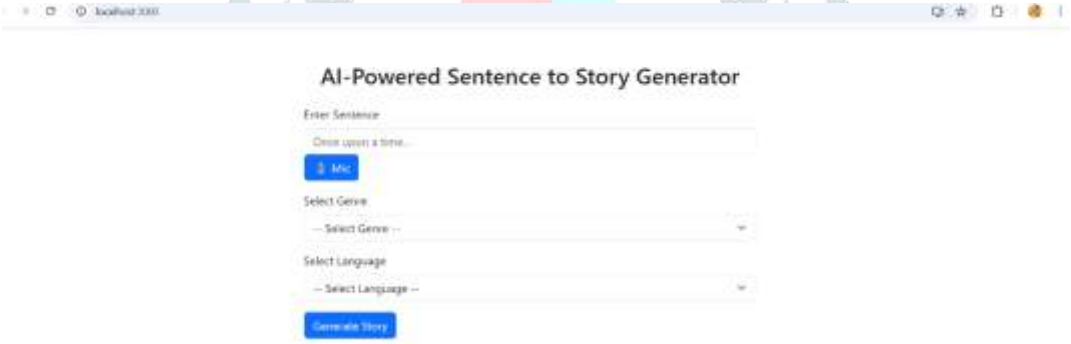
CSS styles for wed app

Mic and speaker modules

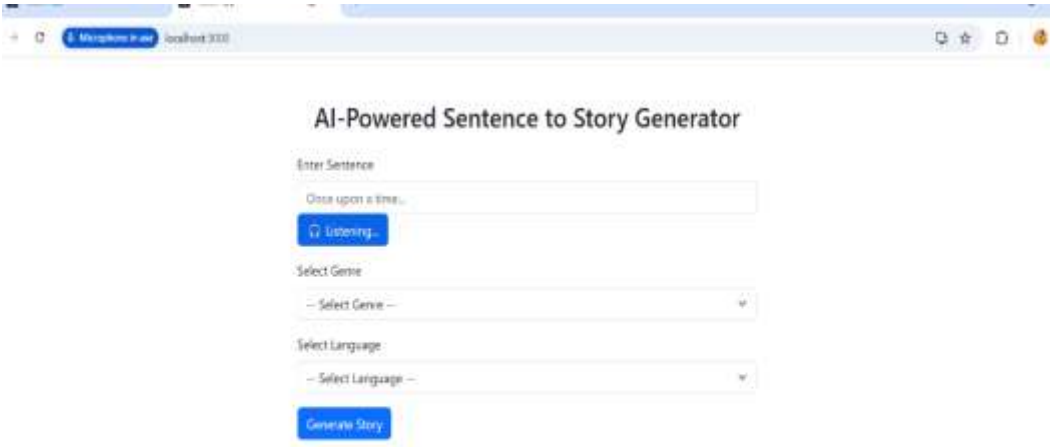
7. RESULT AND ANALYSIS

The proposed AI-managed story generation system was successfully implemented and tested. This allows users to enter a sentence or speak, select a language and style and get a creative story as an output. The system produced meaningful, coherent and style-psychological stories in many languages including English, Telugu, Hindi and French. Using the Deepseek R1 model, the response time to create a 500-word story was less than 5 seconds. Voice input and output increased the user interaction, allowing users to speak both their inputs and listen to the story generated in natural voice. The system was tested on various browsers and equipment and proved to be responsible, reliable and effective in generating suitable stories for educational, entertainment and creative purposes.

Model/Approach	Accuracy (%)	BLEU Score	ROUGE Score	Response Time (sec)	Multilingual Support
Deepseek(R1) through Openrouter	94.2%	0.78	0.84	3.1s	yes
Transformer-based	88.7%	0.63	0.71	4.5s	No
CNN+RNN (Encoder-Decoder)	85.3%	0.60	0.68	4.5s	No



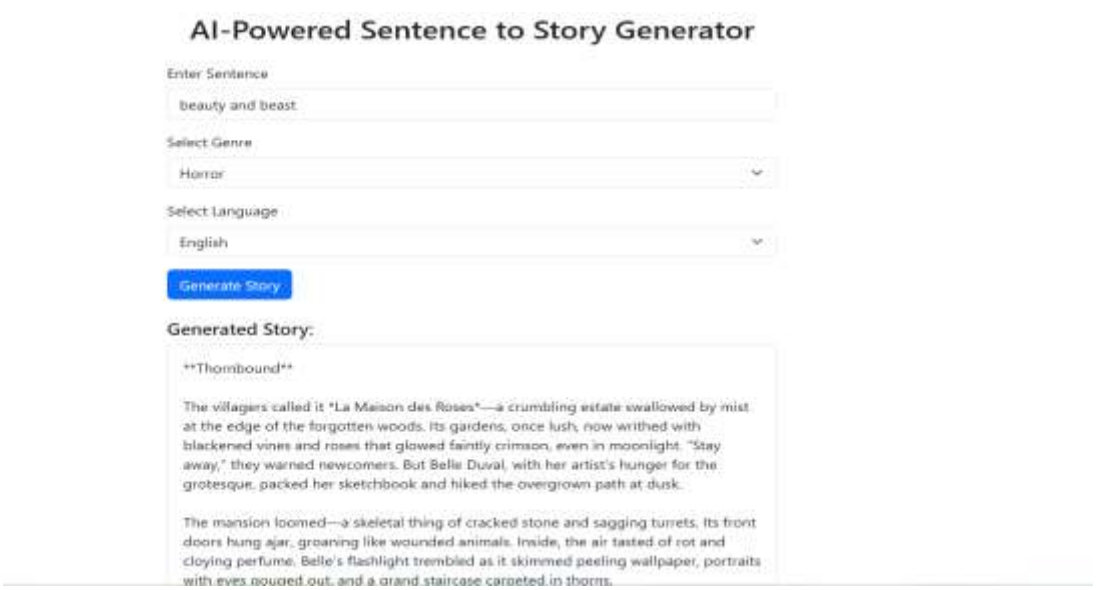
After execute the code the front end



Giving sentence using mic

Enter the all inputs

After enter all the input it generate telugu short story



Generate the English short story

7. CONCLUSION

The development of this multilingual, AI-powered sentence-to-story generator is a significant step forward in combination with artificial intelligence with human creativity. Using the Deepseek R1 via Openrouter, the system is capable of generating, consistent stories enriched by the sentences provided by the simple users in various styles and languages. Integration of speech recognition

and speech synthesis increases further access, allowing users to interact naturally through voice and listen to stories generated in human voice.

The project not only displays the power of modern language models, but also shows practical applications of AI in education, entertainment and creative writing. With its spontaneous interfaces manufactured in react and its real -time performance, this tool serves as a valuable resource for learners, storytellers and developers. Future promotion may include more languages, long -term stories and support for personal storytelling features, its ability and effect may expand. Additionally, introducing interactive story mode, where users can make options and affect the direction of the story, converting the application into a dynamic storytelling game. Advanced TTS features can improve voice output quality such as character voice cloning, multilingual speech synthesis and regional accent support. Developing a mobile application and integrating visual storytelling elements such as AI-based pictures can make the scope and appeal of the system more comprehensive. This enhancement will not only make the application more attractive, but will also be highly impressive in educational, entertainment and access domains.

Acknowledgement

We are thankful to Dr. A. Mary Sowjanya, Department of CS&SE for guiding us, all through the project work.

8. REFERENCES

1. Openrouter ai. (2024). Deepseek R1 model for lesson generation. Https://openrouter.ai
2. Mozilla Developer Network (MDN). (2023). Web Speech API - Speech recognition and synthesis. Https://developer.mozilla.org/en-us/docs/web/api/eb_speech_api
3. React.js documentation. (2024). A JavaScript library for the manufacture of user interfaces. Https://reactjs.org taken from
4. Bootstrap. (2024). Front-end component library for responsible design. Https://getbootstrap.com.
5. IEEE explore. (2022). Storygenai: An automatic style-keyword-based story generation. Https://ieeexplore.ieee.org
6. Suman, R., and Reddy, B.S. (2021). Deep learning-based short story generations for an image using the encoder-decoder structure. International Journal of Advanced Computer Science.