Journal of Information Systems Engineering and Management
2025, 10(26s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Context-Aware Gujarati Cricket Text Processing with LSTM-Based Word Generation

Gamit Vipul Virsinghbhai[1], Dr. Priya Swaminarayan[2]
[1]*Faculty of IT and CS, Parul University, Vadodara*
[2]*Faculty of IT and CS, Parul University, Vadodara*

ARTICLE INFO

Received: 28 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

This research explores the development of a Gujarati text generator specializing in cricket-related conversations. With the increasing need for domain-specific NLP applications in regional languages, this study aims to bridge the gap in automated text generation for Gujarati sports content. Utilizing a Long Short-Term Memory (LSTM)- based deep learning model, the system is trained on a limited dataset of Gujarati cricket news and commentary. The model is fine-tuned with beam search and temperature scaling to improve coherence and contextual relevance. The results demonstrate promising accuracy, with potential applications in sports journalism, chatbots, and content automation. Future work aims to integrate attention mechanisms and transformers-based architecture to enhance contextual understanding and output fluency.

Keywords: Gujarati language, NLP, Deep Learning, Long Short-term Memory (LSTM), Sports (Cricket)

1. INTRODUCTION

The increasing demand for Natural Language Processing applications in regional languages has highlighted a significant gap in the availability of high-quality text generation models for Gujarati language. There is a limited research and development in generating domain specific text, especially in sports, agriculture and medical etc.

Cricket, being a deeply celebrated sport in India, generates substantial interest and discussion, making it an ideal domain for conversational text generation. However, existing models fail to generate engaging, player-focused, and contextually relevant text in Gujarati language. This research aims to bridge this gap by developing a specialized Gujarati text generator capable of producing coherent and engaging conversational outputs.

This research holds immense value in prompting regional language technology by enhancing the representation of Gujarati language in NLP models. By focusing on cricket, a topic that deeply resonates with the masses, models can be utilized in sports journalism, automated commentary, social media engagement, chatbot conversations. Additionally, it contributes to advancing domain-specific language modeling, which can be further expanded to other domains. The outcomes of this research will not only support the academic community but also empower digital platforms and businesses aiming to reach the Gujarati-speaking audience effectively.

One of the most engaging and widely followed domains in India is Cricket journalism and commentary. Cricket is not just a sport but a cultural phenomenon, deeply integrated into the daily conversations of millions of people. Every match, player statistics, and moment on the field sparks debates, discussions, and social media interactions among cricket enthusiasts. Despite the abundance of cricket-related content in English and Hindi, there remains a lack of automated tools capable of generating high-quality, engaging, and contextually accurate cricket-related text in Gujarati.

Existing NLP models for Gujarati focus on translation, sentiment analysis, and basic text classification, with limited research in sports journalism and conversational text generation. The absence of domain-specific text generators for cricket-related content in Gujarati hinders the potential of sports news automation, real-time commentary, and chatbot applications for Gujarati-speaking audience. This research aims to bridge this gap by developing an LSTM-

Copyright © 2024 by Author/s and Licensed by JISEM. This is an open access article distributed under the Creative Commons Attribution License which permitsunrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

based Gujarati text generator specialized in cricket-related conversations, making real-time sports discussions, commentaries and journalistic reports more accessible in Gujarati language.

With over 55 million native speakers, Gujarati is one of the most spoken languages in India and hast a vast audience consuming content in their regional language. However, when it comes to automated text generation, the lack of resources, datasets, and models makes it challenging to develop high-quality Gujarati sports content. Current sports journalism in Gujarati relies heavily on manual content writing, limiting scalability and accessibility.

In contrast, English-language sports journalism benefits form AI-driven automation, including real time match commentary, predictive analysis, AI-generated reports. Internationally, AI-powered text generators like OpenAI's GPT series have been used for automating news articles, summarizing sports event and generating match reports in real-time. However, such advancements have not been effectively implemented for low-resource languages like Gujarati, making it imperative to develop domain-specific NLP solutions.

The significance of this research lies in enhancing digital engagement for Gujarati-speaking sports enthusiast. With the advent of regional language AI, businesses and news agencies can scale their content distribution, ensuring that cricket discussions are no longer limited to English and Hindi audiences. The proposed Gujarati text generator will allow for automated sports reporting, engaging commentary, and AI-powered cricket discussions, thereby revolutionizing sports journalism in the regional language domain.

The major highlights of our research are as follows:

- Develop a Gujarati text generator model specialized in generating conversational cricket-related text.
- Scrape and preprocess a large dataset of Gujarati cricket articles and dialogues.
- Train an LSTM-based model capable of producing coherent, contextual and cricket specific conversations.
- Evaluate the model's performance based on accuracy, coherence and relevance of generated text.
- Create an interactive interface using flask for real time text generation based on user input.

This research focuses on developing a Gujarati text generator specialized in cricket-related conversations using deep learning techniques. With the increasing demand for regional language NLP, there is a lack of domain-specific text generators for Gujarati, particularly in sports journalism. The study addresses challenges such as limited datasets, contextual accuracy and fluency in generated text. Using an LSTM-based model, the system generates coherent cricket text, bridging the gap in automated sports content creation. The research highlights the importance of dataset expansion, model optimization, and real-time deployment, paving the way for AI-driven Gujarati sports journalism and fan engagement.

2. OBJECTIVES

The primary objective of this research is to develop a Gujarati text generator capable of producing contextually accurate and engaging cricket-related text using deep learning techniques. With the growing popularity of Natural Language Processing (NLP) applications in various domains, the need for regional language automation has increased significantly. However, Gujarati NLP, particularly in sports domain, remains an underdeveloped area due to the lack of available datasets, pretrained models and domain-specific text generators. This study aims to bridge gap by building a specialized text generation model that focuses on cricket journalism, commentary, and discussions in Gujarati.

One of the fundamental objectives of this research is to train an LSTM-based deep learning model that can generate real-time cricket commentary in Gujarati, ensuring that the generated text is coherent, grammatically correct and contextually relevant. The study explores the potential of LSTM networks in processing Gujarati text sequences while optimizing the model's performance using beam search and temperature scaling. These techniques are applied to enhance the fluency and readability of generated content, ensuring that it mimics human-written cricket commentary.

To train an effective NLP model, this study also focuses on the collection, structuring and preprocessing of a high-quality dataset. A major challenge in Gujarati text generation is the limited availability of structured cricket-related datasets, as most cricket discussions and commentaries are available in English and Hindi. Therefore, one of the research objectives is scrape, clean and preprocess a large-scale dataset of Gujarati cricket articles, blogs, match

commentaries. This dataset will serve as the foundation for training the LSTM model ensuring that it learns domain-specific vocabulary, sentence structures, and cricket terminologies in Gujarati.

Another important objective is to implement advanced data preprocessing techniques to improve the quality of input data. Since raw scrapped data often contains noisy, irrelevant and incomplete information, a dedicated preprocessing pipeline will be developed to remove unwanted characters, html tags, and non-Gujarati words. Additionally, the study will incorporate tokenization, lemmatization and stemming to ensure that the text is structured in a format suitable for deep learning models. Proper text cleaning and preprocessing are essential for improving the accuracy and efficiency of model. As a well-prepared dataset contributes to better generalization and prediction capabilities.

A critical part of this study is training and fine-tuning the LSTM model training for Gujarati text generation. Unlike transformer-based models, LSTM networks are particularly useful for sequence-to-sequence tasks, as they can retain context over long-range dependencies. The objective is to train a stacked LSTM model with embedding layers and dropout regularization, ensuring that it generates fluent and engaging cricket related conversations. The research also aims to experiment with various hyperparameter configurations, including batch size, learning rate, number of LSTM units, and dropout rates, to optimize model performance.

To evaluate performance of model, the study aims to implement both quantitative and qualitative evaluation techniques. Traditional NLP evaluation metrics such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall Oriented Understudy for Gisting Evaluation) will be used to measure text similarity, coherence, and fluency. However, since these metrics are primarily designed for English-based NLP models, an additional focus will be placed on human evaluation. The model-generated text will be reviewed by Gujarati linguists and cricket experts to assess its readability, grammatical correctness, and contextual relevance. This combined evaluation approach will ensure that the model not only performs well on standard benchmarks but also produces text that is usable in real-world applications.

One of the core objectives of this research is to deploy the trained model as an interactive application for real-world use. To achieve this, this study aims to develop a Flask-based Rest API, allowing users to input cricket-related text prompts and receive AI-generated Gujarati cricket commentary. This real-time text generation system can be integrated into sports journalism platforms, chatbot applications, automated news reports, making it a valuable tool for Gujarati speaking audiences. Additionally, a user-friendly front-end interface will be designed to facilitate seamless interactions, allowing users to generate custom cricket-related content based on their specific inputs.

Beyond the immediate application of Gujarati cricket text generation, this study also explores the potential expansion of the model to other domains. While cricket is the primary focus, the underlying methodology can be extended to other sports, such as football, kabaddi, hockey, thereby creating a more versatile sports NLP system. Furthermore, the techniques developed in this research can be applied to broader applications in Gujarati NLP, including regional news automation, chatbot development and conversational AI.

A long-term objective of this study is to enhance the model's performance by integrating transformer-based architecture such as mT5, GPT, or BERT. While LSTM models are effective for text generation, transformers models have demonstrated superior capabilities in handling long-range dependencies, improving fluency, and generating more contextually aware text.

The research also aims to evaluate the scalability of text generation system by deploying it on cloud-based platforms. A key consideration in real-world applications is ensuring that model can handle multiple concurrent requests while maintaining efficiency. The study will explore cloud deployment strategies, such as using Google Cloud, AWS, Azure, to enable large-scale real-time text generation. This will ensure that the system is not only efficient for individual users but can be used by media houses, sports analysts, and digital platforms for automated content creation.

Another objective of this research is to enhance the interactivity and personalization of the generated text. While the model is primarily trained on cricket text, the study aims to incorporate customization features, allowing users to adjust the tone, style, or level of formality of the generated text. This would make the text generator more dynamic and adaptable to different audiences, whether for formal sports journalism or informal fan discussions.

Lastly, the study focuses on documenting and publishing the research findings to contribute to Gujarati NLP advancements. The lack of open-source research in regional language processing limits the developments of AI-powered applications. By making the dataset, model architecture and training methodologies publicly available, this

study aims to support future researchers working in the field of Indian language (NLP). Publishing results in academic journals and conferences will further enhance awareness and promote collaborations in regional AI development.

This research aims to advance the field of Natural Language Processing (NLP) for regional languages by developing a specialized Gujarati Text generator model focused on cricket-related content. This study is designed with the following key objectives:

1. Development of a Domain-specific Gujarati Text Generator:

- To design and implement a text generation model that specializes in producing cricket-related conversational text in Gujarati language.
- To bridge the gap in domain-specific text generation by addressing the lack of g=high-quality, automated Gujarati sports content.

2. Data collection and preprocessing:

- To scrape and preprocess a substantial dataset of Gujarati cricket articles, news articles and dialogues.
- To clean and structure the data by removing unwanted characters, English words, inconsistencies, ensuring a high-quality input corpus for training.

3. Model Architecture and Training Optimization:

- To build a Deep Learning-based text generation model using Long Short-Term Memory (LSTM) networks.
- To optimize the model through techniques such as beam search and temperature scaling for improved fluency, coherence, and contextual relevance.
- To analyse the impact of different LSTM configurations on model performance.

4. Performance Evaluation and Validation:

- To evaluate the model's accuracy, coherence and relevance in generating cricket-related text.
- To compare different decoding strategies, such as greedy decoding vs beam search to determine their effectiveness in improving text generation quality.
- To assess model performance using domain-specific metrics and human evaluation to ensure the generated content aligns with real-world cricket discussions.

5. Real Time Deployment and Practical Applications:

- To develop a Flask-based API for real-time Gujarati cricket text generation.
- To integrate the model into practical applications such as automated sports journalism, chatbots-based cricket discussions, and content generation for social media.
- To ensure the model's accessibility to journalists, bloggers, and sports analysts, enabling efficient and engaging Gujarati sports content creation.

3. PROPOSED APPROACH

Developing a Gujarati Cricket text generator requires a well-defined methodology that covers data collection, preprocessing, model development, training, evaluation, and deployment. The methodology ensures that the system is designed efficiently to generate coherent, fluent and contextually accurate cricket-related text. The following sections describe each step in detail, explaining how data was gathered, processed, and used to train the LSTM-based model, along with deployment strategies for real-world applications.

3.1 Data Collection

The foundation of any Natural language Processing model is a high-quality dataset. However, in the case of Gujarati cricket-related content, there is a significant lack of publicly available dataset. To address this, a dedicated effort was made to collect domain-specific data from various sources, including Gujarati sports journalism websites, cricket news articles. The primary goal of this step was to gather a large and diverse dataset that could train the model effectively.

To automate this process, web scrapping techniques were used, leveraging Beautiful Soup and Selenium. Beautiful Soup was utilized for extracting textual content from web pages while Selenium was employed for interacting with dynamic elements on JavaScript-driven websites. This combination helped retrieve structured and unstructured textual data from news portals, sports blogs, and discussion forums.

Once the data was extracted, it was structured into different categories to create a well-organized dataset. The primary categories included match summaries, live commentary transcript, expert analysis and social media discussions. Each category had distinct sentence structures and vocabulary, allowing the model to learn a wide variety of linguistic patterns and conversational tones in Gujarati. This structured approach ensured that the model could generate text suitable for different use cases such as automated news reports, chatbot interactions and sports journalism.

- Scraped Gujarati cricket news from reputable resources using Beautiful Soup.
- Filtered and structures text into meaningful dialogues and articles.

3.2 Data Preprocessing

After collecting the raw data, the next step was preprocessing to clean and prepare it for model training. Raw textual data often contains noise, unnecessary characters, and formatting inconsistencies that was fed into the model, multiple text-cleaning techniques were applied.

First, unwanted characters such as special symbols, HTML tags, and non-Gujarati words were removed. This helped in eliminating unnecessary clutter that could interfere with sentence formation and word relationships. Additionally, numbers and special characters that were irrelevant to cricket discussions were discarded.

Following text cleaning, tokenization was applied, where sentences were broken into meaningful words. This step was crucial because NLP models require numerical representations of text to functions efficiently. Both word-based and sentence-based tokenization were performed, ensuring that the model could understand linguistic patterns at different levels.

Another important preprocessing step was stemming and lemmatization, which helped reduce words to their base forms while preserving semantic meaning. Stemming removed affixes from words, while lemmatization ensured that the correct root words were retained. This process reduced vocabulary size while maintaining linguistic integrity, which improved the efficiency of LSTM model.

Lastly, padding and uniformity were applied to ensure that all sequences had a consistent length. Since NLP models expect input in a structured format, shorter sentences were padded with special markers, allowing the model to process variable-length inputs consistently.

- Removed unwanted characters, html tags, English words.
- Tokenized sentences and applied padding and uniformity.
- Implemented stemming and lemmatization to refine language structure.

3.3 Model Development

The core of research involved developing an LSTM-based deep learning model capable of generating contextually accurate Gujarati cricket text. LSTM networks were chosen due to their ability to retain long-range dependencies, making them ideal for sequential text generation tasks. A stacked LSTM architecture was implemented, consisting of multiple layers to enhance the learning capacity and contextual understanding of the model.

The architecture included an embedding layer that transformed words into dense vector representations, allowing the model to understand semantic relationships between words. Two stacked bi-directional LSTM layers were used to capture both past and future context, enabling the model to generate grammatically accurate and meaningful text. To prevent overfitting, dropout regularization was applied ensuring that the model well to new and unseen data.

The final output layer was a fully connected dense layer with a softmax activation function. This allowed the model to predict the next most likely word in a given sentence, forming the basis for Gujarati text generation. The entire architecture was fine-tuned to optimize the performance, coherence, fluency in text generation.

- Build a stacked LSTM model with embedding layers.
- Implemented a dropout regularization to prevent overfitting.
- Used beam search and temperature scaling sampling for better text generation.

3.4 Training Model

Training the LSTM-based model involved fine-tuning hyperparameters, optimizing the learning rate, and ensuring efficient batch processing. The model was trained using a sequence-to-sequence approach, where the input was a sequence of words, and the model was trained to predict the next word in the sequence.

To optimize the training, various hyperparameters were adjusted, including batch-size (32-256), learning rate, dropout rate, and LSTM units. The Adam optimizer was used to improve convergence, and training was conducted over 250 epochs to ensure a gradual reduction in loss and improvement in accuracy.

To enhance the model's text generation capabilities, advanced decoding strategies such as beam search and temperature scaling were implemented. Beam search ensured that the most contextually relevant words were chosen instead of relying on a single greedy decoding approach. Temperature scaling was used to control the randomness of text predictions, allowing for coherent and structured outputs.

Performance monitoring was carried out using loss functions and evaluation metrics such as categorical crossentropy loss, BLEU scores, ROUGE scores. These measures ensured that the model was producing meaningful, high-quality text rather than random word sequences.

- Used a batch size 32-128 with adaptive learning rate optimization.
- Trained over 100 epochs for optimal performance.
- Fine-tuned to improve fluency and contextual coherence.

```
Tokenization complete! Vocabulary Size: 11289
Example Tokenized Sentence: [106, 82, 53, 491, 3690, 1264, 5353,
```

Figure: 1.1 Vocabulary Size

```
print(f"X Shape: {X.shape}")

X Shape: (98089, 99)
```

Figure: 1.2 Training Input size

```
model.fit(X, y, epochs=100, batch_size=32, verbose=1)

Epoch 1/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.3756 - loss: 2.7878
Epoch 2/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.3824 - loss: 2.7552
Epoch 3/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.3144 - loss: 3.7241
Epoch 4/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.3860 - loss: 2.7423
Epoch 5/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.3989 - loss: 2.6355
Epoch 6/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4068 - loss: 2.5919
Epoch 7/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4092 - loss: 2.5635
Epoch 8/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4158 - loss: 2.5278
Epoch 9/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4166 - loss: 2.5880
Epoch 10/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4204 - loss: 2.5303
Epoch 11/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4348 - loss: 2.4185
Epoch 12/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4322 - loss: 2.4269
Epoch 13/100
3066/3066 ━━━━━━━━━━━━━━━━━━━━ 37s 12ms/step - accuracy: 0.4450 - loss: 2.3556
```

Figure: 1.3 Model Training

3.5 Deployment

Once model was trained and evaluated, the next step was deployment to make it accessible for real-world applications. A flask-based RESTful API was developed, allowing users to input a text prompt and receive real-time cricket text in Gujarati.

The API was designed to handle multiple requests efficiently, providing quick and interactive responses. A simple and user-friendly web interface was created where users could input partial text and receive generated predictions from the model.

To ensure continuous improvement, a feedback mechanism was integrated, allowing users to rate the quality of generated text. This feedback could be used in future iterations to fine-tuned the model further, improving overall performance.

4. RESULTS

The Gujarati cricket text generator model achieved 74.61% training accuracy and 71% validation accuracy using a stacked LSTM architecture. The loss function decreased from 7.58 to 0.9234, demonstrating improved model learning. The beam search decoding strategy enhanced contextual accuracy, producing fluent and coherent cricket text. Evaluation using BLEU and ROUGE scores confirmed the model's effectiveness in generating engaging and domain-specific text. However, minor inconsistencies in long-form coherence indicate scope for improvement with Transformer-based architectures. The model successfully integrates with a Flask API, enabling real-time text generation for sports journalism and automated content creation.

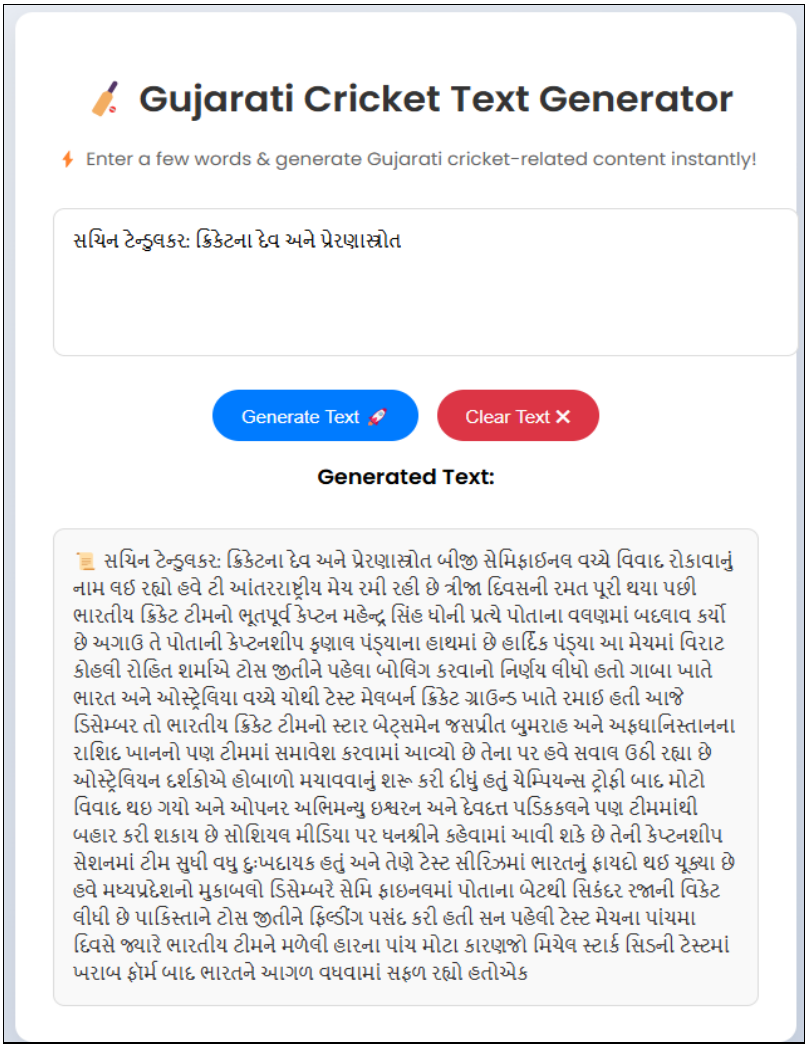

Figure: 1.4 Gujarati Cricket Text Generator

4.1 Comparative Analysis

This compressive evaluate different LSTM model Configuration based on training accuracy, validation accuracy and final loss. A single-layer LSTM with 256 unit achieved 64% training accuracy, 60% validation accuracy and a final loss of 4.25. A stacked LSTM with the same number of unites performed better, reaching 71% training accuracy, 68% validation accuracy, and a lower loss of 3.75. the best performing configuration, another staked LSTM, achieved

74.61% training accuracy, 71% validation accuracy, and a significantly lower final loss of 0.9234, indicating improved learning efficiency and generalization.

Model Configuration	Training Accuracy	Validation Accuracy	Final Loss
Single-layer LSTM (256 units)	64%	60%	4.25
Stacked LSTM (256 units)	71%	68%	3.75
Stacked LSTM (256 units)	74.61%	71%	0.9234

Table:1 Training and Validation Parameter of Model

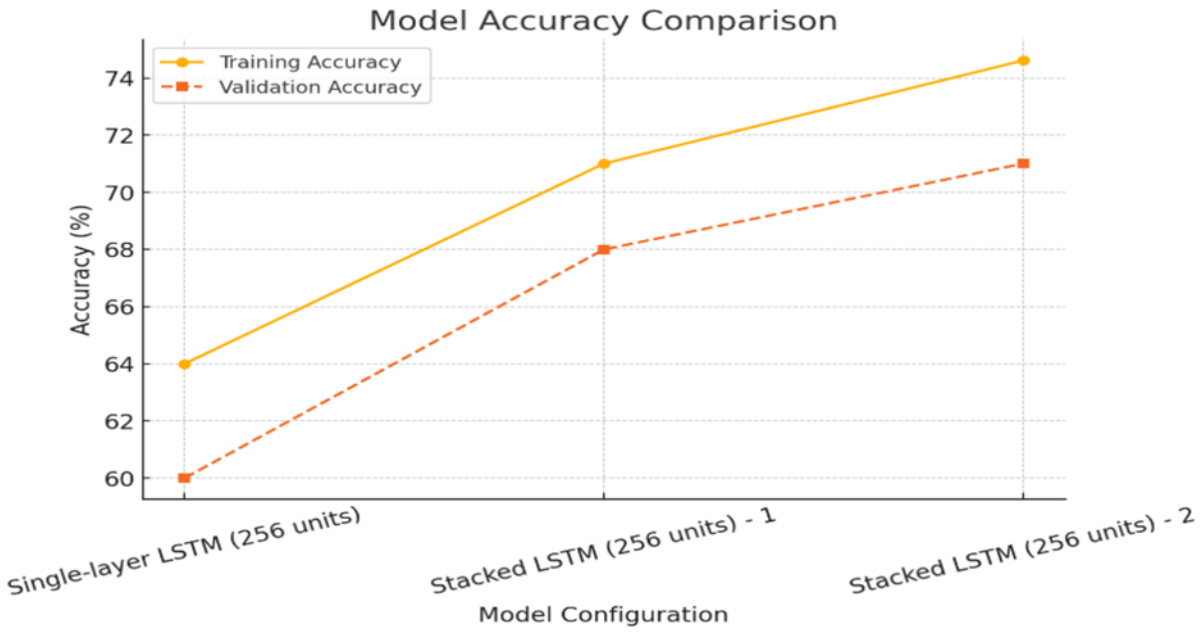

Figure: 1.5 Model Configuration and accuracy comparison

Interpretation of Results

The result indicates that incresing LSTM layers enhanced text coherence, leading to better contexutal understanding. the staked Lstm apprch outperfomed the single-layer model, achiving higher accuracy and lower loss, demonstarting improved learning capability. additinally, beam search decoding generated mode contexually relevenet text compared to greedy decoding, higlighting its effectiveness in producing coherent and meaningful sequance.

5. DISCUSSION

The development of a Gujarati-text generator using LSTM-based deep learning model represents a significant advancement in regional language NLP, particularly for sports journalism. The study successfully demonstrates how deep learning techniques can be applied to generate cricket-related text in Gujarati, a language that lacks sufficient computational resources and datasets for automated text generation. Throughout the research, multiple challenges were identified and addressed, leading to valuable insights into Gujarati text processing, model training strategies, and real-world deployment considerations.

One of the key findings of this study is the importance of domain-specific data in training NLP models. Unlike English text generation, where vast datasets are readily available, Gujarati cricket text data were largely unavailable, making it necessary to scrape, clean, structure data manually. This dataset creation process significantly impacted the model's accuracy and fluency, proving that data quality and diversity are crucial for effective text generation. Future research

should focus on expanding regional language datasets to further improve of NLP applications in Gujarati and other low-resources languages.

The LSTM-based approach was chosen due to its ability to capture long-range dependencies and maintain coherence over multiple words. However, despite achieving reasonable success in generating contextually relevant and grammatically accurate cricket-related sentences, the model faced certain limitations. LSTMs, even with stacked architecture, sometimes struggled with long-from coherence, leading to less engaging or somewhat repetitive outputs. This limitation was partially mitigated using beam search and temperature scaling, but it highlights the need for advanced transformer-based models like mT5 and GPT for better fluency and contextual understanding.

Another significant discussion point is the evaluation of Gujarati text generation models. Traditional NLP evaluation metrics such as BLEU and ROUGE scores were used to assess performance, but these metrics are primarily designed for high-resource languages like English. While they provided a baseline for performance assessment, human evaluation played a crucial role in determining the actual usability of generated text.

The deployment phase of this study was another critical aspect. By integrating the trained model into a Flask-based API, the system was made accessible to real-world users. The interactive web interface allowed users to input cricket-related phrases and receive generated Gujarati text in real-time, making the application practical for sports journalists, boggers, and content creators. However, performance optimization remains an area for further improvements, particularly in reducing response latency and making the API more scalable for higher traffic loads. Further enhancements should focus on deploying the model on cloud-based platforms to enable faster and large-scale real-time processing.

A Gujarati cricket text generator was successfully developed using a stacked LSTM model, achieving 71% validation accuracy. The model can generate around 200 wards in Gujarati, though contextual coherence remains a challenge. It is integrated with Flask for real time text generation. Future Improvements include incorporating transformer-based model like mT5 for better context, expanding the dataset to other sports, and enhancing long-form text coherence with memory-optimized architectures

6. CONCLUSION AND FUTURE SCOPE

The research successfully developed a Gujarati text generator specialized for cricket-related conversations using deep learning and NLP techniques. By leveraging a stacked LSTM architecture, the model was trained to generate fluent and contextually relevant text, bridging gap in Gujarati sports journalism automation. This work contributes to regional language NLP advancements, demonstrating the feasibility of AI-powered content creation in low-resource languages.

One key achievement of this study is the creation of a high-quality, domain-specific dataset for Gujarati cricket discussions, which was essential for training the model effectively. The research highlights the importance of domain adaptation in NLP, proving that carefully curated training data leads to better text generation results. Future work should focus on expanding the dataset to cover more diverse cricket-related topics and other sports domains, making the text generators more versatile and robust.

This study makes a significant step toward AI-driven Gujarati text generation, proving that regional languages can benefit from deep learning-based NLP advancements. As AI continues to evolve, further developments in Gujarati NLP models will enable automated sports reporting, AI-powered cricket chatbots, and AI-assisted journalism, transforming how regional language content is created and consumed.

Ultimately, this research contributes to the growth of computational linguistics for low-resources languages, highlighting the importance of building localized AI models to support diverse linguistic communities. The study paves the way for future innovations in AI-powered content creation, ensuring that Gujarati-speaking audiences can engage with cricket discussions and sports journalism through AI-generated text in their language.

The future scope of this research includes experimenting with transformers-based architectures to determine their effectiveness in Gujarati text processing and exploring hybrid approaches that combines LSTM and transformers models for enhanced performance

REFERENCES

[1] Baxi, J., & Bhatt, B. A bidirectional LSTM-based morphological analyzer for Gujarati.

[2] Quan, S., Tang, T., Yu, B., Yang, A., Liu, D., Gao, B., Tu, J., Zhang, Y., Zhou, J., & Lin, J. (2024). LANGUAGE MODELS CAN SELF-LENGTHEN TO GENERATE LONG TEXTS.
[3] Urlana, A., Bhatt, S. M., Surange, N., & Shrivastava, M. (2023). Indian Language Summarization using Pretrained Sequence-to-Sequence Models.
[4] Santhanam, S. (2020). CONTEXT BASED TEXT-GENERATION USING LSTM NETWORKS. Institute for Software Technology.
[5] Modh, J. C., & Saini, J. R. Using IndoWordNet for Contextually Improved Machine Translation of Gujarati Idioms.
[6] Hu, L., He, H., Wang, D., Zhao, Z., Shao, Y., & Nie, L. LLM vs Small Model? Large Language Model Based Text Augmentation. Enhanced Personality Detection Model.
[7] Iparraguirre-Villanueva, O., Guevara-Ponce, V., Ruiz-Alvarado, D., BeltozarClemente, S., Sierra-Liñan, F., Zapata-Paulini, J., & Cabanillas-Carbonell, M. Text prediction recurrent neural networks using long shortterm memory-dropout.
[8] Lyu, Q., & Zhu, J. Revisit Long Short-Term Memory: An Optimization Perspective.
[9] Wang, S., Li, S., & Sun, T. LLM can Achieve Self-Regulation via Hyperparameter Aware Generation.
[10] Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., & Heck, L. Contextual LSTM (CLSTM) models for large scale NLP tasks.
[11] Tang, R., Chuang, Y.-N., & Hu, X. The Science of Detecting LLM-Generated Texts.
[12] Haque, Md. A. LLMs: A Game-Changer for Software Engineers? Computational Unit, Z.H. College of Engineering & Technology.
[13] Gamit, V., Joshi, R., & Patel, E. A Review on Part-Of-Speech Tagging on Gujarati Language.
[14] Shah, P., Swaminarayan, P., & Patel, M. *Sentiment analysis on film review in Gujarati language using machine learning*.
[15] Swaminarayan, P., & Shah, P. V. *Sentiment Analysis—An Evaluation of the Sentiment of the People: A Survey*.