Grenze International Journal of Engineering and Technology, January Issue

Design of a Spell-Checker and Grammar-Checker for Kannada Documents

RJ Prathibha[1], Ananya M[2], Dhanya V Hegde[3] and Dhanika Gowda D[4]
[1-4] JSS Science and Technology University, Mysore, India
Email: rjprathibha@sjce.ac.in@sjce.ac.in, {ananyamkoundinya, dhanyavhegde01, dhanikagowda222}@gmail.com

***Abstract*—This paper presents a new Grammatical Error Correction (GEC) tool for the Kannada language that combines rule-based and machine learning approaches. The spell-checker makes use of Hunspell library and custom dictionary consisting of 4789 words, resulting in an accuracy of 80%. The machine learning approach for error detection uses the pre-trained DistilBERT model, which is trained using a custom dataset of 45798 sentences, achieving an accuracy of 92%. The rule-based approach for error correction is able to achieve an accuracy of 90%. The performance of the GEC tool shows that it can significantly improve the quality of written Kannada.**

I. INTRODUCTION

Natural Language Processing (NLP) combines linguistics, computer science, and artificial intelligence to enable machines to understand and produce human language. Among its various applications, grammatical error correction stands out with the important task of identifying and correcting grammatical errors in a text. This study introduces a new GEC model for the Kannada language spoken by millions in South India. Despite the growing importance of NLP, there is a significant gap in language tools for many Indian languages, including Kannada. Our research addresses this gap by developing a comprehensive GEC tool that combines both rule-based and machine learning approaches.

II. RELATED WORKS

Recent NLP research has made significant progress in grammatical error correction and spell-checking in various languages. In particular, research has focused on solving the problems of highly inflexible Indian languages and other complex language systems. In Indian languages, an Long Short Term Memory(LSTM)-based approach to OCR detection and error correction has shown high F-scores and reduced word errors in Sanskrit, Malayalam, Kannada, and Hindi[1]. In Urdu, a hybrid model combining Soundex and Shapex algorithms and N-gram and LCS techniques achieved an F1 metric of 98.68% for the best proposals[2].
In "A Context-Sensitive Real-Time Spell Checker with Language Adaptability" (2020)[3] by Prabhakar Gupta, the central focus is on devising a spell checker capable of functioning in real-time and adjusting to diverse languages. The study adopts a comprehensive methodology, leveraging a combination of Wikipedia articles and carefully curated video subtitles to construct dictionaries tailored for different languages. This approach facilitates the system's ability to efficiently detect and rectify spelling errors as they occur. Evaluation of the system reveals impressive precision rates across multiple languages, showcasing its superiority over established tools such as GNU Aspell and Hunspell.

Grenze ID: 01.GIJET.11.1.43

However, the research also acknowledges challenges associated with memory requirements, particularly concerning the storage of extensive dictionaries, and variations in performance stemming from differences in dataset quality. Despite these hurdles, the study underscores the promising potential of the proposed approach in advancing spell-checking capabilities across a range of linguistic contexts. "Real-word Errors in Arabic Texts: A Better Algorithm for Detection and Correction" (2019)[4] by Aqil M. Azmi, Manal N. Almutery, and Hatim A. Aboalsamh addresses the challenge of identifying and rectifying real-word errors in Arabic texts. Their Java-based system leverages publicly available modules and libraries for text preprocessing, incorporating stemmers like Khoja, Light10, and MADAMIRA.

Using the Damerau-Levenshtein distance for error generation and DALM for language modeling, alongside SVM models generated using Weka, the system achieves high accuracy rates, particularly with MADAMIRA stemmer and non-linear SVM kernels like RBF and sigmoid.

Despite promising results, limitations include a decline in performance with increased distinct words in a corpus and occasional failure to select the most appropriate correction candidate. Overall, the study offers a comprehensive approach to real-word error detection and correction in Arabic texts, with the potential for further refinement in the error correction mechanism to enhance real-world applicability.

"A rule-based Bengali grammar checker" (2021)[5] by A.N.M Fahim Faisal, Md. Afikur Rahman, and Tanjila Farah introduces a novel system addressing grammatical errors in Bengali text, filling a gap in existing research focused on spell checking, Text to Speech (TTS), and Optical Character Recognition (OCR). Utilizing a dataset of Bengali sentences, the study trained and validated the system by tokenizing and tagging sentences with Parts of Speech(POS) using the BNLP toolkit.

The system then matched these POS-tagged sentences against predefined Bengali grammar rules, achieving promising accuracy rates of 92% for Set 1, 80% for Set 2, with an average of 86%. However, the reliance on rudimentary POS tags and predefined rules may lead to occasional false positives or negatives, highlighting the need for further refinement to handle complex sentence structures with greater precision for broader applicability. A Python-based system using NLP libraries has shown 94% accuracy in English spell-checking[6]. A hybrid approach for Sinhala combining rule-based methods with machine learning achieved 88.6% accuracy[7]. Advanced models of neural networks, especially transformers, have shown potential in online grammar-checking systems[8].

Parochial training with pre-trained BERT embeddings improved grammatical error detection in grammars[9]. "Symspell and LSTM-based spellcheckers for Tamil" (2020) by Selvakumar Murugan, Tamil Arasan Bakthavatchalam, and Malaikannan Sankarasubbu presents a thorough investigation into spell-checking methods designed for the Tamil language, emphasizing the challenges specific to Indian languages.

The study evaluates three spell checkers: bloom-filter, symspell, and an LSTM-based approach, utilizing two primary datasets. While the bloom filter proved simplistic yet effective for basic spell checking, symspell offered fast correction suggestions at the expense of significant memory requirements. The LSTM-based approach, though innovative, demonstrated limited practicality with an exact match score of only 0.40 and high computational demands.

The study discusses the practical implications and limitations of these approaches across different runtime environments, emphasizing the need for further research to develop more robust spell-checking tools for Tamil[10]. Taken together, these studies show the diversity of approaches and ongoing challenges in developing effective GEC and spell-checking tools for different languages. They emphasize the need for language-specific solutions and the potential of hybrid and neural-network based approaches.

III. METHODOLOGY

The Kannada GEC system has been developed using a modular architecture with four main components as shown in Fig 1, viz:

- User input module
- Spell-checking module
- Grammatical error detection module
- Grammatical error correction module

The detailed description about each module is given below.

A. User input module

This module provides an interface for users to input Kannada text for analysis and correction.

B. Spell-checking module

This uses a comprehensive Kannada dictionary consisting of 4879 words. It checks each word in the dictionary for spelling errors, using the Levenshtein distance algorithm to generate repair suggestions based on similarity measures.

C. Grammatical error detection module

This performs syntactic analysis of input sentences. It focuses on identifying subject-verb agreement errors and other grammatical inconsistencies specific to Kannada.

D. Grammatical error correction module

This implements a rule-based approach to error correction. It uses language rules and patterns adapted to Kannada grammar. It fixes subject-verb agreement problems, tense mismatches, and other syntax errors, and creates corrected versions of the input statements.

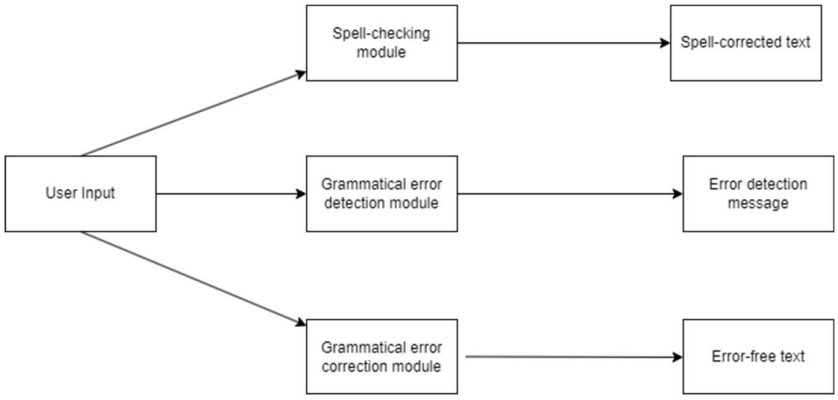

Fig. 1. High-level Design of GEC System

IV. PROPOSED SYSTEM

The proposed GEC system has three main components: spell-check and correction, grammatical error detection and grammatical error correction.

A. Spell-check and Correction

The system uses the Hunspell library to check and correct spelling. This makes use of a Kannada-specific dictionary and affix files (Fig. 2). The process involves:

- Tagging the input text
- Checking each word from Hunspell dictionary
- Proposing corrections for misspelled words
- Text reconstruction with corrections

B. Grammatical Error Detection

This module uses machine learning using DistilBERT(Fig. 3). This process includes the following:

- The Kannada corpus is pre-processed and labelled.
- The data is divided into training, testing, and validation sets.
- The DistilBERT model is loaded and refined on Kannada dataset.
- Model performance is evaluated using test data.
- The trained model is frozen for deployment.

C. Grammatical Error Correction

A rule-based approach is applied focusing on subject-verb agreement errors(Fig. 4). This involves the following:

- Input sentences are spell-checked.
- Kannada-specific vocabulary helps to identify parts-of-speech.
- Grammar rules coded into the system check subject-verb agreement.
- The system suggests corrections based on these rules.

This hybrid approach combines the efficiency of rule-based systems with the learning ability of neural networks and handles both spelling and grammatical errors in the text. The system's modular design enables independent

optimisation of each component while maintaining overall integration for comprehensive text analysis and correction.

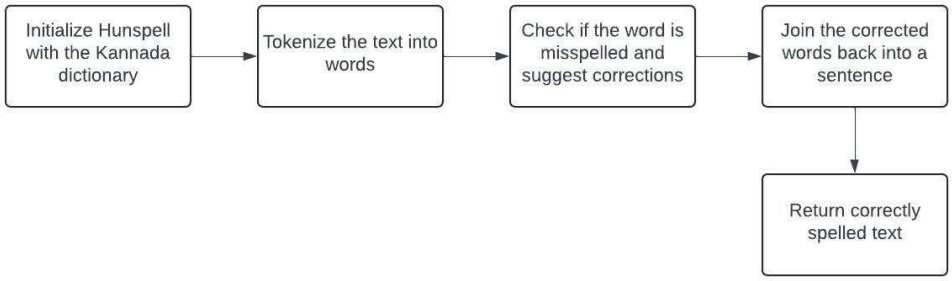

Fig. 2. Spell-check and Correction

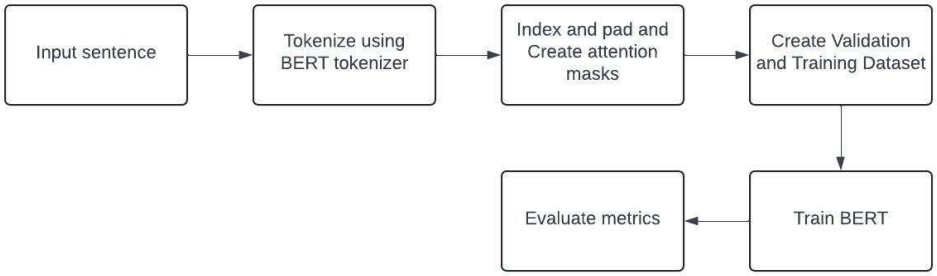

Fig. 3. Grammatical Error Detection

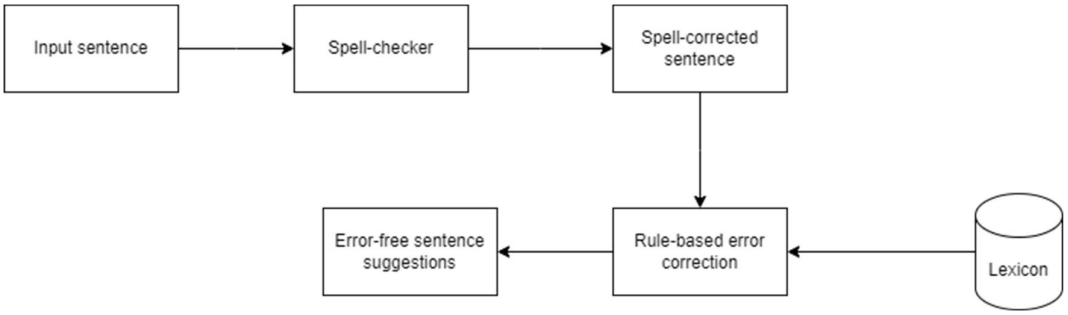

Fig. 4. Grammatical Error Correction

V. IMPLEMENTATION

The Kannada GEC system comprises three main modules: spell-checking and correction, grammatical error detection, and grammatical error correction.

A. Spell-check and Correction

The system utilizes the Hunspell library, an open-source tool designed for languages with complex morphology. A custom Kannada lexicon containing 4,879 commonly used words was developed. The module employs Levenshtein distance to suggest corrections for misspelt words, leveraging Hunspell's dictionary (.dic) and affix (.aff) files for morphological analysis.

B. Grammatical Error Detection

This module implements the DistilBERT model, a compact version of BERT (Bidirectional Encoder Representations from Transformers). The model was fine-tuned on a custom dataset of 45,798 Kannada

sentences in present and future tenses, labelled for grammatical correctness. DistilBERT is a Transformer model that is compact, lightweight, quick, inexpensive, and trained by distilling BERT basis. Compared to google-BERT/BERT-base-uncased, it uses 40% fewer arguments and executes 60% quicker, all while maintaining over 95% of BERT's performance as determined by the GLUE language comprehension benchmark. With comparable performance, DistilBERT seeks to offer a version of BERT that is more computationally efficient. This is accomplished by using a method known as "knowledge distillation," in which a smaller model is trained to emulate the actions of a bigger model. DistilBERT, on the other hand, has a much lower parameter set and a quicker inference time while learning to replicate the representations produced by the original BERT model.

C. Grammatical Error Correction

A rule-based approach was implemented, focusing on subject-verb agreement errors in Kannada. The system identifies subjects and verbs through stemming and comparison against a curated list of common Kannada nouns and verbs. It then suggests verb corrections based on the subject's count, gender, and person, particularly for present and future tenses.

An example illustrating subject-verb agreement in Kannada is shown in Fig. 6. In this case, verb "hōguttāne" agrees with singular subject "avanu," while verb "hōguttāre" corresponds to plural subject "avaru.".

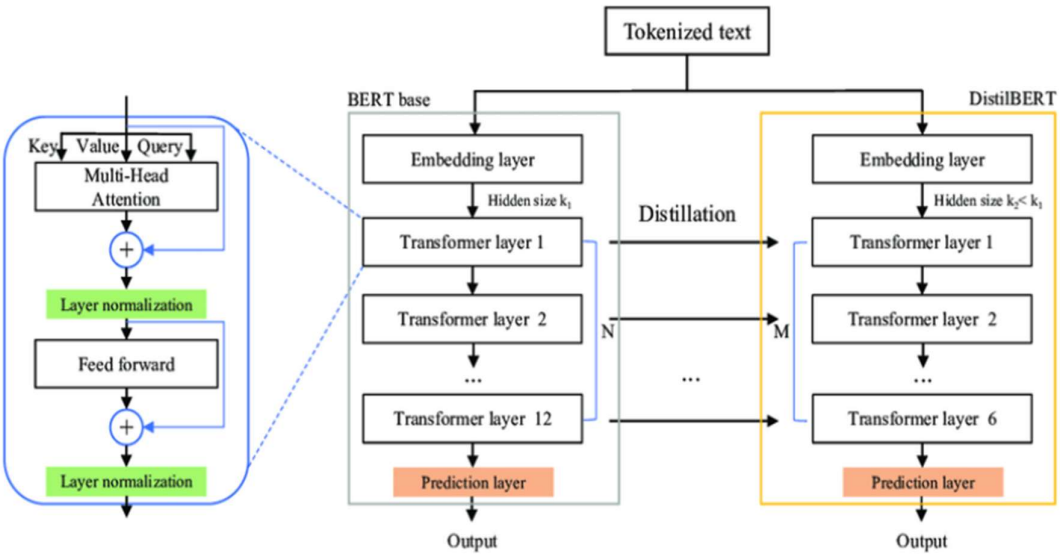

Fig. 5. DistilBERT model

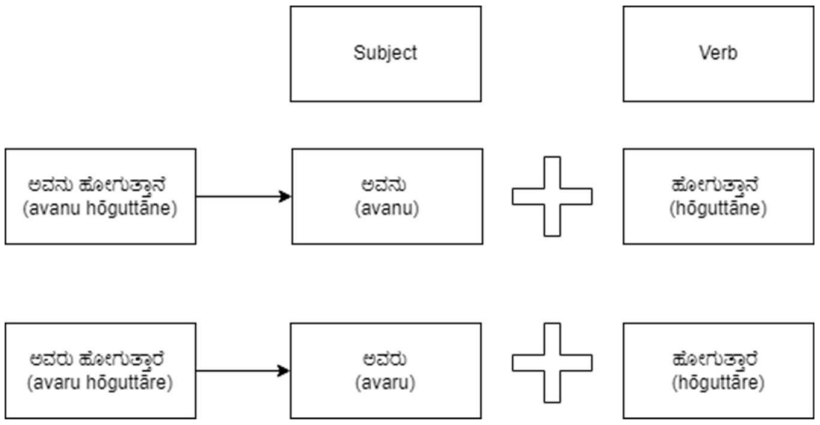

Fig. 6. Subject-verb Agreement

VI. RESULTS

A. Spell-check and Correction

The Kannada spell-checker underwent testing with a sample of 20 randomly selected words containing errors. Out of these, expected corrections were provided for 16 words.

Accuracy = No. of incorrect words corrected by spell-checker/Total no.of incorrect words provided (1)

The accuracy achieved by the proposed system as per the above equation is 80%. This indicates that the spell-checker module has successfully identified and rectified errors in most of the tested Kannada words. Fig. 7 and Fig. 8 demonstrate the working of spell-checker.

B. Grammatical Error Detection

The error-detection model was trained using DistilBERT. The training data comprised two categories of sentences, those with grammatical errors and those without grammatical errors. Fig. 9 the classification report of the model that was trained to detect grammatical errors in Kannada text. The test set included a mix of sentences with and without grammatical errors. The model was able to achieve an accuracy of 92% on the test set containing 5088 sentences.

C. Grammatical Error Correction

The Kannada GEC module was tested with a sample of 20 randomly chosen sentences in present and future tense, containing errors. Out of these, expected corrections were provided for 18 sentences.

Accuracy = No.of incorrect sentences corrected by grammar-checker/Total no.of incorrect sentences provided

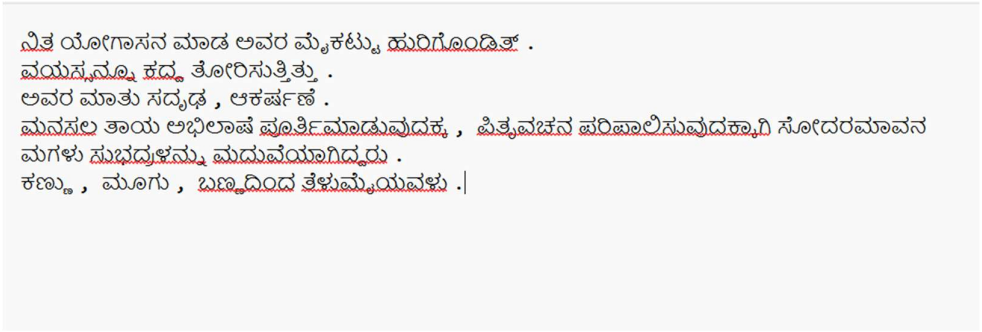

Fig. 7. Input of Spell-checker

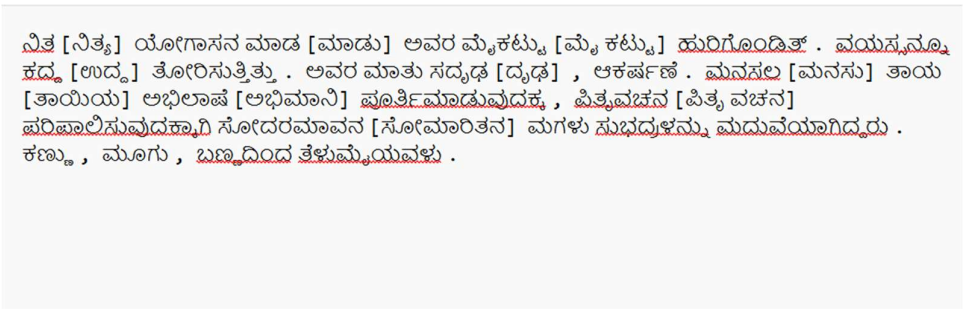

Fig. 8. Output of Spell-checker

	precision	recall	f1-score	support
0	0.80	0.44	0.56	539
1	0.93	0.99	0.96	4041
accuracy			0.92	4580
macro avg	0.87	0.71	0.76	4580
weighted avg	0.91	0.92	0.91	4580

Fig. 9. Classification Report

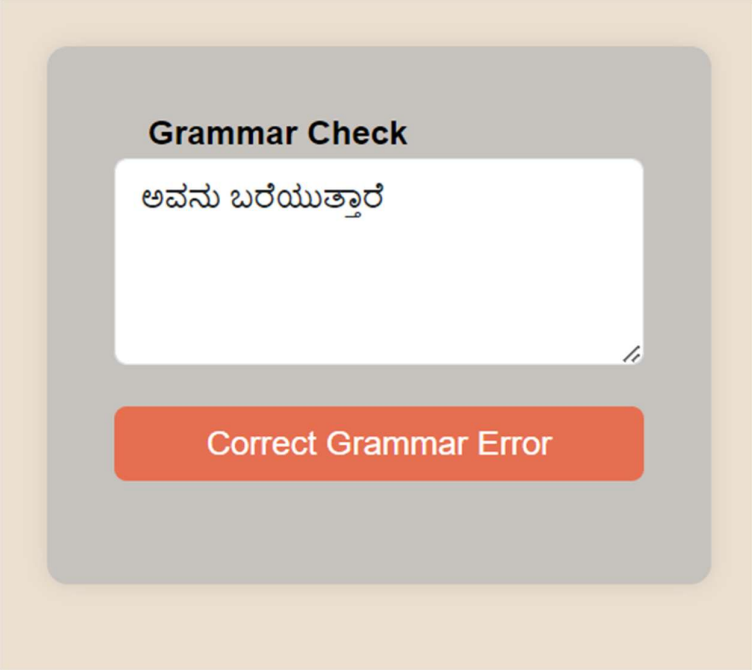

Fig. 10. Input for Grammatical Error Correction

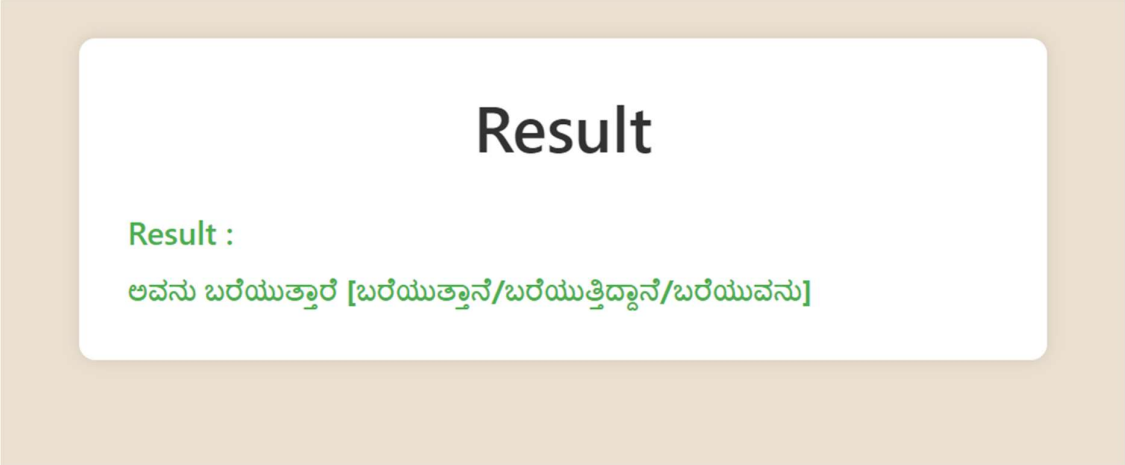

Fig. 11. Output of Grammatical Error Correction

VII. Conclusion

The main goal of this project is to create a spelling and GEC tool for Kannada texts using the BERT model and to assess how well it works. In the past, Kannada GEC has been done using LSTM technology. In this project, we are trying a different approach using Transfer Learning. Additionally, we have added a spell-checking feature to further improve the accuracy of the tool. This project aims to make the GEC process more effective and user-friendly for Kannada documents.

Future Work

As we move on with our work, we will be concentrating on improving the spell and grammar-checking tool to handle irregular verb usage, varied tenses, and intricate sentence structures. In order to efficiently parse and evaluate complex sentences, our next phase of research will focus on refining algorithms and linguistic models, taking into account the dynamic nature of language and the subtleties inherent in written communication. We hope to give users more thorough
feedback on sentence structure by integrating sophisticated syntactic and semantic parsing tools, guaranteeing coherence and clarity in their writing. Moreover, broadening the system's scope to include additional irregular verbs and tenses will improve the system's accuracy and applicability in many language contexts.
Through constant research, testing and iterative improvements, we are committed to making the spell and grammar-checking tool a versatile and indispensable tool for writers who want to improve their skills and communicate with precision and finesse.

References

[1] Saluja, Rohit, Devaraj Adiga, Parag Chaudhuri, Ganesh Ramakrishnan and Mark James Carman. “Error Detection and Corrections in Indic OCR Using LSTMs.” 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) 01 (2017): 17-22.

[2] Aziz, Romila & Anwar, Muhammad & Jamal, Muhammad Hasan & Bajwa, Usama. (2021). A hybrid model for spelling error detection and correction for Urdu language. Neural Computing and Applications. 33. 10.1007/s00521-021-06110-7.

[3] Gupta, Prabhakar. (2020). A Context-Sensitive Real-Time Spell Checker with Language Adaptability. 116-122. 10.1109/ICSC.2020.00023.

[4] Azmi, Aqil & Almutery, Manal & Aboalsamh, Hatim. (2019). Real-Word Errors in Arabic Texts: A Better Algorithm for Detection and Correction. IEEE/ACM Transactions on Audio, Speech, and Language Processing. PP. 1-1. 10.1109/TASLP.2019.2918404.

[5] Faisal, A. & Rahman, Afikur & Farah, Tanjila. (2021). A Rule-Based Bengali Grammar Checker. 113-117. 10.1109/WorldS451998.2021.9514031.

[6] Gondaliya, Yash & Kalariya, Poojan & Panchal, Brijeshkumar & Nayak, Amit. (2022). A Rule-Based Grammar and Spell Checking. SAMRIDDHI A Journal of Physical Sciences Engineering and Technology. 14. 48-54. 10.18090/samriddhi.v14i01.8.

[7] H. M. U. Pabasara and S. Jayalal, "Grammatical error detection and correction model for Sinhala language sentences," 2020 International Research Conference on Smart Computing and Systems Engineering (SCSE), Colombo, Sri Lanka, 2020, pp. 17-24, doi: 10.1109/SCSE49731.2020.9313051.

[8] Hao, Senyue & Hao, Gang. (2020). A Research on Online Grammar Checker System Based on Neural Network Model. Journal of Physics: Conference Series. 1651. 012135. 10.1088/1742-6596/1651/1/012135.

[9] Wang, Quanbin & Tan, Ying. (2020). Grammatical Error Detection with Self Attention by Pairwise Training. 1-7. 10.1109/IJCNN48605.2020.9206715.

[10] Murugan, Selvakumar & Bakthavatchalam, Tamil Arasan & Sankarasubbu, Malaikannan. (2020). SymSpell and LSTM based Spell- Checkers for Tamil.