International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 11 | Nov 2026 www.irjet.net p-ISSN: 2395-0072

Marathi Grammar and Spelling Checker Using Natural Language Processing

Rutuja Gondikar[1], Pooja Dane[2], Nikita Fandade[3]

[123]*Department of Computer Science Engineering, International Centre of Excellence in Engineering and Management, Chhatrapati Sambhajinagar, Maharashtra, India*
Project Guide: Prof. Anjali Hiwale | Head of Department: Prof. Sujata Chechre

---

Abstract - *Marathi is widely used in education, administration, journalism, and everyday digital communication. However, spelling mistakes and grammatical inconsistencies can reduce the clarity of Marathi text. This paper presents a Python-based prototype for Marathi text correction using a two-stage Natural Language Processing (NLP) pipeline. In the first stage, the Spello SpellCorrectionModel configured for Marathi is trained on a small sentence corpus and used to suggest spelling corrections. In the second stage, the spelling-corrected text is passed to the multilingual mBART-50 many-to-many machine translation checkpoint to generate Marathi output. The implementation is organized into three modules: MarathiSpellChecker, MarathiGrammarChecker, and MarathiLanguageChecker. The combined module retains the original input, intermediate spelling-corrected text, generated output, and spelling correction metadata. The system is intended as an educational prototype and an initial framework for experimentation rather than a validated production grammar checker. The spelling corpus is small, and the selected mBART-50 checkpoint is designed for machine translation; it has not been fine-tuned on Marathi grammatical-error-correction pairs in this implementation. Therefore, no accuracy or performance claim is made. The paper defines a reproducible evaluation procedure using manually reviewed Marathi test sentences, exact-match accuracy, Word Error Rate, and human assessment. The work highlights the need for larger Marathi datasets, task-specific model training, and meaning-preservation checks.*

Keywords - Marathi NLP, spelling correction, grammatical error correction, Spello, mBART-50, Python, Devanagari

1. INTRODUCTION

Natural Language Processing (NLP) enables computers to analyze, process, and generate human language. Applications include machine translation, text classification, information extraction, speech interfaces, and writing assistance. A spelling and grammar checker is a writing-support application that identifies likely language errors and proposes alternatives.

Marathi is an Indo-Aryan language written primarily in the Devanagari script. Accurate Marathi writing involves spelling conventions, morphology, agreement, word order, tense, aspect, and context. A spelling checker generally focuses on word-level errors, whereas grammatical correction may require understanding relationships between words and rewriting a phrase or sentence.

This work describes a two-stage Marathi text-processing prototype. The first stage uses Spello with the language code 'mr' to suggest spelling corrections. The second stage uses the checkpoint facebook/mbart-large-50-many-to-many-mmt through Hugging Face Transformers to generate Marathi output. The model is configured with the Marathi language token mr_IN. This is an experimental generation step; because the selected checkpoint is a machine-translation model and is not fine-tuned for Marathi grammatical error correction in the supplied implementation, its output cannot automatically be treated as a verified grammatical correction.

1.1 Problem Statement

The project addresses the design of a Python-based prototype that accepts Marathi text, applies spelling correction, and then performs a grammar-oriented generation step. The system should preserve the original sentence, the spelling-corrected intermediate text, the final generated text, and spelling correction metadata so that each processing stage can be examined.

1.2 Objectives

- To develop a modular Python application for Marathi spelling and grammar-oriented text processing.
- To configure Spello for Marathi spelling correction using a sentence corpus.
- To load mBART-50 through Hugging Face Transformers and generate Marathi output.
- To integrate both stages and retain intermediate and final outputs.
- To define a repeatable test procedure and identify limitations for future improvement.

2. RELATED WORK AND BACKGROUND

Spell checking commonly involves error detection and candidate generation. Dictionary-based systems compare tokens with known vocabulary, while edit-distance methods estimate the number of insertions, deletions, or substitutions needed to transform one word into another. Statistical and neural approaches can use word frequency and context to rank alternatives. Such methods may struggle with proper nouns, regional vocabulary, inflected forms, and words missing from the training corpus.

Grammatical Error Correction (GEC) differs from spelling correction because an error may involve agreement, tense, word order, missing words, redundant words, or sentence construction. Neural sequence-to-sequence systems can generate a complete corrected sentence, but reliable GEC generally requires suitable pairs of erroneous and corrected sentences for training or fine-tuning.

The Transformer architecture introduced attention-based sequence processing and has become central to modern language models [1]. mBART is a multilingual sequence-to-sequence model based on denoising pre-training [2]. The selected mBART-50 many-to-many checkpoint is documented for multilingual machine translation [3]. Setting a target language token to Marathi guides the generated language, but it does not change the model's training objective into grammatical error correction.

The proposed prototype combines an Indian-language spelling component with multilingual text generation. Its contribution is primarily an educational integration and an inspectable processing workflow. It does not claim to introduce a new Marathi language model or to outperform existing correction systems.

3. PROPOSED SYSTEM

3.1 System Architecture

The system follows a sequential pipeline: (1) Marathi text input, (2) spelling correction using Spello, (3) extraction of corrected text and correction dictionary, (4) Marathi text generation using mBART-50, and (5) return of a structured result. The result contains four fields: original, spell_corrected, grammar_corrected, and errors.

Stage	Component	Output
Input	User-provided Marathi sentence	Original text
Spelling correction	Spello SpellCorrectionModel(language='mr')	Corrected text and correction dictionary
Generation	mBART-50 with Marathi language token	Generated Marathi text
Integration	MarathiLanguageChecker	Combined result dictionary

3.2 Module Design

MarathiSpellChecker initializes SpellCorrectionModel(language='mr'), trains it using a supplied or default Marathi sentence corpus, and invokes spell_correct(text). The implementation configures minimum and maximum spelling-correction token lengths as 2 and 25. The returned spell_corrected_text and correction_dict fields are extracted from the package result. Exact package behavior should be checked against the installed Spello version.

MarathiGrammarChecker loads MBart50TokenizerFast and MBartForConditionalGeneration from Hugging Face Transformers. It sets tokenizer.src_lang to 'mr_IN', tokenizes the input, and generates output with the Marathi language token as the forced beginning-of-sequence token. The supplied configuration uses beam search with num_beams=5 and max_length=256. Generation is performed under torch.no_grad() and the decoded sequence is returned.

MarathiLanguageChecker coordinates the two modules. Its check(text) method first stores the original text, then applies spelling correction, extracts the intermediate text and spelling correction dictionary, and passes the intermediate text to the generation module. The method returns the original, spell_corrected, grammar_corrected, and errors fields. The model objects are initialized once and reused for multiple inputs.

3.3 Processing Algorithm

- Initialize and train the Marathi Spello model using the available sentence corpus.
- Load the mBART-50 tokenizer and model; configure the Marathi language token.
- Accept a Marathi input sentence.
- Apply spelling correction and extract the corrected text and correction metadata.
- Pass the spelling-corrected text to the mBART-50 generation stage.
- Decode the generated sequence and return all stages in a structured result.

3.4 Example Test Inputs

Test input	Purpose / intended check
मला मराठी भाशा आवडते	Check whether the spelling 'भाशा' is corrected to 'भाषा'.
आज हवामान खुप छान आहे	Check whether 'खुप' is corrected to 'खूप'.
तो खूप बोलतो आहे	Inspect grammar-oriented generation; validate naturalness with a Marathi reviewer.
मला मराठी भाषा आवडते	Check whether already-correct text is unnecessarily changed.

These are illustrative test inputs and intended checks, not measured outputs. The actual output must be recorded after running the program in the target environment.

4. IMPLEMENTATION AND EVALUATION METHODOLOGY

4.1 Software and Hardware Requirements

The implementation uses Python 3.x, Spello, Hugging Face Transformers, and PyTorch. A modern multi-core processor and sufficient memory are required. The project report recommends 16 GB RAM because the mBART-large checkpoint is memory intensive; a compatible CUDA-capable GPU is optional. Several gigabytes of storage and internet access may be required for the initial model download. Exact software versions and machine specifications should be recorded during testing.

4.2 Evaluation Procedure

The system should be evaluated on a manually reviewed Marathi test set that is separate from the small corpus used to train the spelling component. The same test cases should be processed using the original input, Spello-only output, and combined pipeline output. Reference corrections should be prepared before testing and reviewed by a fluent Marathi speaker. Raw outputs should be saved exactly as generated.

4.3 Evaluation Metrics

- Exact-match accuracy: (number of outputs exactly matching the approved reference / total test cases) × 100.
- Word Error Rate (WER): (substitutions + deletions + insertions) / number of words in the reference.

- Human review: classify each output as correct and meaning-preserving, partially improved, unchanged despite an error, over-corrected, meaning altered, or worse than the input.

WER measures textual difference and does not by itself establish grammatical quality or semantic preservation. Exact-match accuracy is strict and may count acceptable alternative phrasings as mismatches. Metrics should therefore be reported alongside human review.

4.4 Results Status

No quantitative experimental results are reported in this manuscript because the supplied project material does not contain completed execution logs or verified output measurements. Accuracy, runtime, and error-rate values must be added only after the program has been executed and the outputs have been reviewed. The test environment should record the operating system, Python version, Spello version, Transformers version, PyTorch version, processor, RAM, GPU/CUDA details, model checkpoint, test-set size, and experiment date.

5. DISCUSSION AND LIMITATIONS

The modular design makes it possible to inspect the spelling and generation stages separately. This is useful for debugging and for identifying whether an error originates in spelling correction or subsequent generation. However, errors may compound: the spelling module may replace a valid word, and the generation model may then produce a sentence based on the altered input.

The current spelling corpus contains only a small number of Marathi sentences and is not representative of the language's vocabulary, regional forms, or morphology. The mBART-50 checkpoint is designed for machine translation and has not been fine-tuned on paired Marathi grammatical-error-correction examples in this implementation. Consequently, the generation stage may leave an error unchanged, paraphrase the input, or alter its meaning.

Other limitations include the absence of confidence scores, correction explanations, accept/reject controls, a user interface, and systematic handling of model download failures or memory constraints. Proper nouns, dialect forms, punctuation, and already-correct sentences require explicit testing. The system should therefore be treated as a prototype and its suggestions should be reviewed by users.

6. CONCLUSION AND FUTURE SCOPE

This paper presented a Python-based prototype for Marathi spelling correction followed by experimental Marathi text generation. Spello is used for the spelling stage, while the mBART-50 multilingual checkpoint generates Marathi output. The integrated class retains the original input, intermediate spelling result, final generated text, and spelling correction metadata.

The work demonstrates a modular NLP workflow, but it does not establish validated grammatical correction performance. A model's ability to generate text in Marathi does not prove that it can detect and correct Marathi grammar. Future work should expand the corpus, prepare a manually reviewed parallel dataset of erroneous and corrected Marathi sentences, fine-tune or select a task-specific GEC model, and evaluate the system on held-out examples. A user interface with correction highlighting, explanations, and accept/reject controls could also improve usability.

Potential applications include educational writing support, proofreading of Marathi notices and reports, and assistance for local-language content creators. These applications should be considered only after correction quality, meaning preservation, and user experience have been evaluated.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Project Guide Prof. Anjali Hiwale for guidance and support during the project. The authors also acknowledge Prof. Sujata Chechre, Head of the Department, Department of Computer Science Engineering, International Centre of Excellence in Engineering and Management, for academic support and encouragement. The authors

thank the developers and maintainers of Python, Spello, Hugging Face Transformers, PyTorch, and the mBART model for the software and research resources used in this prototype.

REFERENCES

[1] A. Vaswani et al., "Attention Is All You Need," in Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS), 2017.

[2] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual Denoising Pre-training for Neural Machine Translation," Transactions of the Association for Computational Linguistics, vol. 8, pp. 726–742, 2020.

[3] Hugging Face, "mBART-50 Many-to-Many Multilingual Machine Translation Model," model documentation. Available: https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt.

[4] Hugging Face, "Transformers Documentation." Available: https://huggingface.co/docs/transformers/.

[5] Spello project documentation and source repository, "Spello: Spell Correction for Indian Languages." Verify the exact repository URL and installed version before submission.

[6] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. draft. Available: https://web.stanford.edu/~jurafsky/slp3/.

[7] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," Soviet Physics Doklady, vol. 10, pp. 707–710, 1966.

[8] Python Software Foundation, "Python 3 Documentation." Available: https://docs.python.org/3/.

[9] PyTorch, "PyTorch Documentation." Available: https://pytorch.org/docs/.