Grammar Feedback for Non-Native Hindi Learners

Aditya Shah
Dept. of Computer Science
George Mason University
ashah49@gmu.edu

Nikhil Chukka
Dept. of Computer Science
George Mason University
nchukka@gmu.edu

Uddip Yalamanchili
Dept. of Computer Science
George Mason University
uyalaman@gmu.edu

1 Introduction

1.1 Task / Research Question Description

The objective of this research is to develop a personalized grammar feedback tool for non-native Hindi learners. Addressing the lack of effective resources to identify and correct Hindi grammatical errors, a key challenge for self-directed learners. By creating models to detect and correct these errors, this study aims to support learners in achieving fluency.

1.2 Motivation and Limitations of existing work

Previous efforts in grammar correction for non-native learners have primarily focused on widely spoken languages such as English, with limited resources for Hindi, particularly personalized feedback. Existing tools often lack accuracy and adaptability to Hindi's complex grammar, focusing more on vocabulary or basic structures.

Our approach addresses these gaps by targeting Hindi grammatical errors using datasets such as Wikipedia edit histories and synthetic error generation. Advanced NLP models such as DistilBERT, T5, and MarianMT are fine-tuned for Hindi's unique grammatical features, tackling token-level inconsistencies and syntactic and morphological complexity.

By integrating real-world error datasets and focusing on sentence-level correction, this research aims to deliver accurate and effective tools for Hindi grammar correction.

1.3 Proposed Approach

The proposed approach aims to develop a system for real-time feedback on grammatical errors in Hindi, using advanced NLP models such as T5, MarianMT, and DistilBERT, which are tailored for Hindi grammar correction.

We utilized a mix of synthetic error datasets and real-world data from Wikipedia edits to train these models, focusing on sentence-level correction to address the limitations of token-level approaches in handling Hindi's complex grammar. Experiments with various architectures helped identify the models that are best suited to their syntactic and morphological richness.

This approach combines grammar detection with correction, allowing learners to understand and rectify their mistakes effectively.

1.4 Likely challenges and mitigations

One of the key challenges is sourcing a high-quality dataset containing both grammatically correct and incorrect sentences. To address this, we have developed a synthetically generated dataset designed for classifying sentences based on grammatical understanding. Other challenges include accurately detecting complex grammatical errors, handling diverse sentence structures, and ensuring the tool's performance. To mitigate these, we used transformer models as they can understand the context and grammar in a sentence.

2 Related Work

2.1 Generating Inflectional Errors for Grammatical Error Correction in Hindi

The challenge of grammatical error correction (GEC) in Hindi, a morphologically complex and low-resource language, has been addressed by creating a synthetic corpus with inflectional errors and a natural error corpus

from Wikipedia edits. Using a modified ERRANT toolkit, the analysis revealed that inflectional errors are the most prevalent. This dual dataset approach facilitates the evaluation of Hindi GEC models, using state-of-the-art methods originally developed for English. The study shows that artificial datasets effectively train models for inflectional errors, but incorporating broader error types and manual curation could further improve system performance, offering a foundation for similar research in other Indic languages. (Sonawane et al., 2020)

2.2 Vyakranly: Hindi Grammar & Spelling Errors Detection and Correction System:

Vyakranly is an automated tool for Hindi that detects and corrects both spelling and grammar errors using a combination of rule-based and statistical methods, including morphological analysis and part-of-speech tagging. It is designed for simple sentences but can handle some compound structures, distinguishing itself by integrating grammar checking, spell correction, and translation between Hindi and English (no actual implementation). However, the model struggles with complex sentence patterns and may produce less accurate results when dealing with intricate grammatical structures or idiomatic expressions.(S. et al., 2023)

2.3 Detection and Correction of Grammatical Errors in Hindi Language Using Hybrid Approach:

This article describes a Hindi grammar-checking system created with a hybrid approach that incorporates statistical and rule-based approaches. The approach efficiently corrects four common grammatical problems, including number and gender-related adjective-noun and noun-verb agreement difficulties. It uses a combination of morphological analysis, part-of-speech tagging, and pattern-based bigram and trigram models to detect and fix problems in short Hindi phrases. The system produced strong performance measures, including an accuracy of 0.83, recall of 0.91, and F-measure of 0.87. However, its architecture is largely focused on basic sentence constructions, restricting its ability to handle more complex grammatical types. (Mittal et al., 2019)

2.4 Frequency-based Spell Checking and Rule-based Grammar Checking :

In this paper, authors developed a hybrid system combining frequency-based spell checking with rule-based grammar checking, focusing on implementing comprehensive tense rules using JSON for rule representation and demonstrating effective encoding of grammatical rules through POS tagging and parsing. (Singh et al., 2016)

2.5 Bangla Grammatical Error Detection Using T5 Transformer Model:

The Bangla Grammatical Error Detection paper (2023) showcased the application of T5 Transformer (60M parameters) fine-tuned on 9,385 sentences with error symbols, achieving a Levenshtein Distance of 1.0394 through extensive post-processing and specialized correction mechanisms for morphologically rich language features. (Shahgir and Sayeed, 2023)

Our approach targets Hindi grammatical errors using datasets like Wikipedia edit histories and synthetic error generation. We fine-tuned advanced NLP models (DistilBERT, T5, MarianMT) to handle Hindi's grammatical complexities, focusing on sentence-level correction and real-world error datasets to provide accurate tools for Hindi grammar correction.

3 Methodology

The development was approached in multiple stages, targeting effective, real-time grammar corrections in Hindi. This section outlines the methodology, including data preparation, model selection, training, and evaluation. We have provided below detailed descriptions of the procedures and techniques, drawing from established practices while incorporating novel adaptations for Hindi grammar.

3.1 Data Collection and Preprocessing

Our approach begins with dataset preparation. We created two key datasets: a synthetic error corpus and a real error corpus. The

synthetic error corpus was generated by introducing common grammatical mistakes into correct sentences, leveraging a rule-based approach on the Hindi Wikipedia Dump (using WikiExtractor and Hindi POS tagging). The real error corpus, named HiWikEd, was built from edit histories on Hindi Wikipedia, extracting sentence pairs before and after corrections. Both datasets were split into training, validation, and test sets to ensure robust evaluation. (Sonawane et al., 2020)

3.2 Model Selection and Training

For the grammar detection models, we initially used BERT(Devlin et al., 2019) for checking if a given sentence is grammatically incorrect or not, and then for token-level error classification, but later transitioned to sequence-to-sequence models for sentence-level correction due to limitations with tokenization. We fine-tuned the T5 model(Raffel et al., 2023) for correcting grammatical errors using the synthetic datasets. We also trained MarianMT (Junczys-Dowmunt et al., 2018) for grammar correction, leveraging its pre-existing translation abilities and fine-tuning it with correct-incorrect sentence pairs to boost its adaptability to Hindi grammar.

We used GPU-accelerated training with mixed precision (Micikevicius et al., 2018) for optimizing performance on large datasets. The Hugging Face Trainer API (Tra) was utilized for training both models, with batch processing and gradient accumulation to accommodate memory constraints. We incorporated early stopping and validation-based model selection to prevent overfitting and to ensure reliable performance.

3.3 Evaluation

To evaluate our system, we have employed a combination of automated metrics to ensure a comprehensive assessment. Automated metrics such as precision, recall, F1-score, BLEU, and GLEU (Mutton et al., 2007) will measure the system's accuracy and fluency in correcting grammatical errors. Unit tests were implemented to validate individual components, including data preprocessing, model training, and inference, ensuring the system's robustness and correctness. In addition to automated evaluation, human assessment was conducted to evaluate the naturalness, usability, and overall quality of the corrected sentences, addressing aspects that automated metrics did not fully capture. This dual evaluation approach provided a thorough understanding of the system's performance and its effectiveness in real-world scenarios.

4 Experiments

4.1 Datasets

Please find the link for the dataset here.

4.1.1 Synthetic Error Corpus

This dataset was generated by introducing grammatical errors into correct Hindi sentences using a rule-based approach. The Hindi Wikipedia dump [1] (June 1, 2024) served as the data source, processed with WikiExtractor (Sonawane et al., 2020) to extract and clean plain text. Errors were introduced focusing on inflectional aspects of verbs, adverbs, and pronouns, leveraging POS tagging for accurate error generation. After discarding 40% of the initial sentence pairs, the remaining dataset was split into training and validation sets in an 80:20 ratio.

4.1.2 HiWikEd Corpus

This dataset comprises real grammatical corrections extracted from Hindi Wikipedia edits. A modified version of WikiEdits(Sonawane et al., 2020) was used to effectively handle Hindi text, extracting sentence-level edits from the revision histories of the Hindi Wikipedia Revision Dump (June 1, 2024). Filters based on sentence length, edit distance, and content were applied to remove noisy edits.

4.1.3 Word Ordering Errors

A tool was developed using PyTorch (Paszke et al., 2019) tensors to introduce word order errors by shuffling sentence structures. The tool utilizes GPU for optimized performance, batch processing, and parallelization for scalability. Sentences are processed incrementally to manage memory constraints. The augmented dataset was specifically created for this research and is not publicly accessible.

[1] https://dumps.wikimedia.org/hiwiki/

4.1.4 Number Agreement and Case Marker Errors

Errors related to number agreement and case markers were introduced into Hindi sentences for NLP tasks using Stanza (Qi et al., 2020). Stanza's tokenization and POS tagging capabilities were used to modify nouns, adjectives, and case markers. Batch processing and GPU acceleration ensured efficient error generation and storage management. The augmented dataset was specifically created for this research and is not publicly accessible.

4.1.5 Ordered Back-Translation

Data augmentation was performed using ordered back-translation, translating sentences from Hindi to English and back to Hindi via Helsinki-NLP translation models (Tiedemann and Thottingal, 2020). The Hugging Face Transformers library (Wolf et al., 2020) enabled batch processing with GPU acceleration for faster translation. This process generated variations of sentences for testing on smaller datasets. The augmented dataset was specifically created for this research and is not publicly accessible.

4.2 Implementation

In this project[2], we experimented with three primary models: DistilBERT (Sanh et al., 2020) for sentence-level correctness checking, BERT for token-level correction, and MarianMT along with T5 Transformer for sentence-level grammar correction in Hindi.

Initially, we aimed to determine if a given sentence was grammatically correct or incorrect using DistilBERT. This approach allowed us to classify whether the sentence conformed to grammatical norms but lacked the ability to provide precise corrections. We then attempted token-level correction using BERT, which encountered significant challenges such as fragmented tokenization of Hindi words and limited contextual understanding.

To resolve these issues, we moved to sentence-level correction using MarianMT and T5 Transformer models, which effectively maintained the context while making corrections, leading to improved accuracy. We used the Hugging Face Transformers library, PyTorch for model training, and employed BLEU and GLEU scores to evaluate model performance.

The implementation details for the models are as follows:

- **DistilBERT for Sentence-Level Classification:** We employed "distilbert-base-multilingual-cased" (Sanh et al., 2020) model to classify sentences as grammatically correct or incorrect.
- **BERT for Token-Level Correction:** (Devlin et al., 2019) The pre-trained "bert-base-multilingual-cased" model was used to attempt token-level corrections. However, it was insufficient for the requirements of complete and contextual grammatical correction.
- **MarianMT and T5 for Sentence-Level Correction:** The "Helsinki-NLP/opus-mt-hi-en" (Tiedemann and Thottingal, 2020) and "t5-small" models were used for sentence-level correction. The MarianMT model (Junczys-Dowmunt et al., 2018) was initially used in its baseline form and subsequently fine-tuned for improved correction capabilities.

Throughout this journey, we tested various hyperparameter settings, used gradient accumulation to manage GPU memory effectively, and performed multiple fine-tuning runs to enhance model performance.

4.3 Results

We compared the results obtained from the different approaches used in our experiments.

Table 1 shows how a fine-tuned model compares to a baseline distilBERT model using precision, recall, and F1-score. The fine-tuned model consistently performs better, especially on the test set, where it shows a noticeable boost in precision, recall, and F1-score.

Table 2 presents a comparison of the BLEU and GLEU scores for both the baseline and fine-tuned models.

The MarianMT model achieved the highest scores, suggesting that sequence-to-sequence

[2]Github Project Link

Model	Validation Set			Test Set		
Metric	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Baseline Model	0.71	0.70	0.70	0.48	0.36	0.41
Fine-Tuned Model	0.84	0.81	0.83	0.68	0.58	0.62

Table 1: Comparison of distilBERT Performance Metrics on Validation and Test Sets for Baseline and Fine-Tuned Model

Model Type	BLEU Score	GLEU Score
Baseline MarianMT Model	0.52	0.43
Fine-Tuned MarianMT Model	0.57	0.49
Change (%)	+9.6%	+13.9%

Table 2: Comparison of BLEU and GLEU scores for baseline and fine-tuned model

models are better suited for the grammar correction task than token-level models. The T5 model also demonstrated a considerable improvement over the baseline with a significant increase in both BLEU and GLEU scores. However, the BERT model's performance was limited, as it could only flag tokens as correct or incorrect without correcting the sentence.

An example of model performance before and after fine-tuning is shown below:

- **Incorrect Sentence:** "उसकी प्रतिभा की गहराई किसी अनजाने समुद्र जैसा है।"
- **Baseline MarianMT Output:** "उसकी प्रतिभा की गहराई किसी अनजाने नदी जैसा है।"
- **Fine-Tuned MarianMT Output:** "उसकी प्रतिभा की गहराई किसी अनजाने समुद्र जैसी है।"

This example illustrates how the fine-tuned model correctly addressed grammatical agreement, which the baseline model failed to do. The fine-tuned T5 model also demonstrated similar improvements in maintaining proper noun-verb agreements in translated sentences.

4.4 Discussion

During the development of the grammar correction model, several challenges emerged, necessitating changes in approach.

- **Initial Sentence-Level Classification:** DistilBERT was effective for identifying whether sentences were grammatically correct. However, it lacked the capability to provide corrective feedback.
- **Token-Level Correction with BERT:** The attempt to perform token-level correction using BERT was largely unsuccessful due to various issues:
  - **Fragmentation of Tokens:** Hindi words were frequently split into subtokens, making it difficult to identify or suggest corrections for entire words effectively. For example, the word "गलत" was split into tokens "ग" and "लत" , making contextual correction inaccurate.
  - **Contextual Limitations:** Token-level corrections did not adequately consider the broader sentence context, leading to incoherent corrections. For instance, the BERT model suggested corrections for individual tokens but failed to provide meaningful suggestions when analyzing the entire sentence structure.
- **Switch to Sentence-Level Correction:** Given the shortcomings of the token-level approach, we shifted our focus to sentence-level correction using MarianMT and T5 models. These models were pre-trained for sequence-to-sequence tasks, making them inherently more capable of handling the complexities of full-sentence corrections while maintaining contextual integrity.
- **Training Challenges:** The experiments were computationally intensive, particularly when fine-tuning the models. The training was performed on a single A100

80GB GPU. To mitigate out-of-memory errors, we adjusted batch sizes, used gradient accumulation, and explored mixed-precision training where possible. Each model took approximately 12-16 hours to train.

- **Sensitivity Analysis:** Runs with different random seeds produced consistent results, demonstrating the robustness of the fine-tuned models in providing reliable corrections. Notably, the MarianMT model achieved a significant increase in BLEU and GLEU scores following fine-tuning.

The transition to using T5 and MarianMT provided a more robust and effective approach to correcting grammatical errors in Hindi sentences. The ability to consider the sentence as a whole allowed these models to make coherent corrections, which was not achievable with token-level classification.

4.5 Resources

The experimentation required considerable computational resources, including training on a single A100 80GB GPU. Each experiment took approximately 12-16 hours. The primary human effort involved:

- Tuning hyperparameters to manage memory constraints and optimizing training performance.
- Troubleshooting challenges during token-level correction, which ultimately led to the decision to switch to sentence-level correction.
- Developing appropriate evaluation metrics (BLEU and GLEU) to assess model performance in grammatical correction tasks.

Training MarianMT and T5 models involved iterative testing and refinement, especially during fine-tuning. Collaboration with peers, discussions on parameter settings, and shared observations contributed significantly to overcoming challenges, particularly those related to memory and resource management.

4.6 Error Analysis

We performed an error analysis to better understand the shortcomings of our models.

MarianMT Model: The MarianMT model, after fine-tuning, was effective in most simple sentence correction tasks. However, there were still some areas where the model exhibited errors:

- **Idiomatic Phrases:** The model often struggled with idiomatic phrases and frequently translated them literally, resulting in loss of intended meaning. For example:
  - **Incorrect Sentence:** "आँख का तारा होना"
  - **Fine-Tuned Output:** "Be the apple of the eye" (Literal translation without cultural context).
- **Complex Sentences:** Nested and compound sentences posed challenges for the model. It was observed that the model produced partially corrected sentences that still contained subtle grammatical errors. For instance:
  - **Incorrect Sentence:** "अगर वह आता तो हम जाते।"
  - **Fine-Tuned Output:** "अगर वह आ जाता हम जाते।" (Grammatical structure not fully corrected).

BERT Token-Level Correction: Token-level correction with BERT faced several specific challenges:

- **Fragmentation of Tokens:** Hindi words were often split into sub-components, making it challenging for BERT to provide coherent corrections. For instance, the word "गलत" was often split, resulting in misleading token labels.
- **Incorrect Labels for Complex Grammar:** BERT struggled with correctly labeling tokens when dealing with intricate sentence structures. It could not effectively capture the dependencies necessary for handling compound or idiomatic expressions.

These observations justified the decision to employ sequence-to-sequence models such as MarianMT and T5, which inherently provided better results in maintaining sentence integrity and correcting grammatical errors effectively.

5 Conclusion

We explored multiple approaches for Hindi grammar correction, starting with sentence-level classification using DistilBERT to detect grammatical correctness. As token-level correction faced limitations, we moved to sentence-level correction using models like MarianMT and T5.

Through experiments, fine-tuning these models showed significant improvements over non-fine-tuned versions, as evidenced by increased BLEU and GLEU scores. Despite notable gains in accuracy and language understanding, challenges remain in handling complex grammatical structures. Future work will focus on refining the models, enhancing robustness, and further optimizing the correction capabilities for complex sentences.

References

Trainer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. ArXiv:1810.04805.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. ArXiv:1804.00344 [cs].

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed Precision Training. ArXiv:1710.03740 [cs].

M. Mittal, S. K. Sharma, and A. Sethi. 2019. Detection and Correction of Grammatical Errors in Hindi Language Using Hybrid Approach. 7(5):421–426.

Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. 2007. GLEU: Automatic Evaluation of Sentence-Level Fluency. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 344–351, Prague, Czech Republic. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. ArXiv:1912.01703 [cs].

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. ArXiv:2003.07082 [cs].

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. ArXiv:1910.10683 [cs].

Rachel S., Vasudha S., Shriya T., Rhutuja K., and Lakshmi Gadhikar. 2023. Vyakranly : Hindi Grammar & Spelling Errors Detection and Correction System. In *2023 5th Biennial International Conference on Nascent Technologies in Engineering (ICNTE)*, pages 1–6.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv:1910.01108 [cs].

HAZ Shahgir and Khondker Salman Sayeed. 2023. Bangla grammatical error detection using t5 transformer model. *arXiv preprint arXiv:2303.10612*.

Shashi Pal Singh, Ajai Kumar, Lenali Singh, Mahesh Bhargava, Kritika Goyal, and Bhanu Sharma. 2016. Frequency based spell checking and rule based grammar checking. In *2016 international conference on electrical, electronics, and optimization techniques (iceeot)*, pages 4435–4439. IEEE.

Ankur Sonawane, Sujeet Kumar Vishwakarma, Bhavana Srivastava, and Anil Kumar Singh. 2020. Generating Inflectional Errors for Grammatical Error Correction in Hindi. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 165–171, Suzhou, China. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – Building open translation services for the World. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.