Automatic spell and grammar correction using NLU in Indian Languages

Gireesh Kumar Gopakumar
(*Artificial Intelligence and machine learning)*
Ramaiah university of applied Sciences
Peenya,India
gireeshkumar055@gmail.com

Dr Nayana B R
(*Artificial Intelligence and machine learning)*
Ramaiah university of applied Sciences
Peenya,India
nayana.cs.et@msruas.ac.in

***Abstract*— This research explores NLU-driven spell and grammar correction in Indian languages, addressing linguistic complexities. Customized transformer-based models are developed and evaluated across diverse contexts. Results showcase NLU's potential in enhancing language correction tools, benefiting linguists and tech developers, fostering nuanced digital communication.**

I. Introduction

Natural Language Understanding (NLU) is a critical AI discipline enabling machines to comprehend human language intricacies. By analyzing context, syntax, and semantics in text or speech, NLU empowers machines to accurately derive meaning. This capability forms the foundation for various applications like chatbots, sentiment analysis, and language translation. NLU's significance lies in enhancing human-computer interaction and automating context-aware language tasks. In the realm of error and grammar correction in Indian languages, NLU plays a pivotal role in achieving precise and contextually relevant results. NLU-driven systems excel in rectifying errors while preserving intended meaning by addressing morphological variations, syntactic structures, and contextual nuances. While English has seen progress, the complexities of Indian languages pose unique challenges that require tailored solutions. NLU-equipped error correction tools bridge the gap between human communication subtleties and automated correction, fostering effective communication in languages with intricate grammatical structures. Grammatical Error Correction (GEC) automatically rectifies written text errors, encompassing spelling, syntax, and coherence. Modern approaches leverage statistical and deep learning methods, diverging from traditional rule-based methods. GEC treats errors as a translation task, transforming improperly constructed sentences into grammatically correct forms within the same language. Successful GEC relies on annotated data containing sentence pairs of erroneous and corrected versions. However, research has primarily focused on English and a few well-resourced languages, neglecting languages like Indic. Systems like UTTAM and SCMIL address spelling correction in Indic languages, while simpler models handle common errors..

II. Related Work

Grammatical Error Correction (GEC) datasets are primarily sourced from annotated language learner essays, often originating from tasks like Helping Our Own, CoNLL2014, and BEA2019 for English and corresponding corpora for languages like Russian and Czech. However, generating annotated datasets for low-resource languages proves challenging due to their scarcity and resource-intensive creation process. An alternative approach involves injecting errors into grammatically correct sentences, either through rule-based systems or strategies like round-trip translation. This approach, while effective for languages with limited training data, relies on the accuracy of artificially generated errors mirroring real-world mistakes to create reliable training sets. Another approach involves mining edits from websites like language learner platforms, Wikipedia, and GitHub. Although this method yields natural datasets, the lack of human curation introduces noise, as not all text corrections are grammatical; some aim to improve semantics or provide additional information. The choice of dataset generation method should consider language intricacies, grammatical error annotation needs, and desired dataset quality. Ultimately, the chosen approach should align with the specific language scenario and resource availability to ensure effective training and evaluation of GEC models.

III. Language Selection

Hindi, recognized as a fusional language, exhibits a distinctive linguistic mechanism by which it imparts intricate grammatical attributes like case, gender, number, and tense through morphological alterations. This phenomenon is most pronounced in the realm of verbs and select adjectives, which undergo inflection to harmonize with the number and gender of their associated nouns. This pattern extends to encompass genitive pronouns, post-positions, and ordinals as well. Additionally, verbs exhibit shifts in accordance with the speaker's designated person, while adjectives exhibit shifts in response to the grammatical case of the accompanying noun.

A noteworthy characteristic of Hindi is the reliance on vowel endings situated to the right of the lexical base. These endings serve as markers for the various grammatical changes mentioned earlier. This linguistic framework is aptly demonstrated through examples provided in Table 1, illustrating how these morphological shifts manifest within the language's constructs. In instances where the appropriate inflection is not applied, the resultant sentence tends to appear ungrammatical, primarily due to a breakdown in the expected agreement between different linguistic elements.

Sentence	Complexity
"राम ने खाना खाया।" (Ram ate food.)	Involves subject-object-verb order, verb conjugation
"मुझे किताब दी गई है।" (I was given a book.)	Depicts passive voice, tense agreement

Table 1: Language complexity example

This intricate system of morphological modifications underscores the importance of recognizing the specific patterns and structures that underlie Hindi's grammatical framework. Mastery of these intricate adjustments is essential for effective communication in Hindi, as the language heavily relies on these mechanisms to convey nuances and nuances and maintain grammatical consistency. A deeper understanding of these features not only enriches one's grasp of the language but also ensures a more precise and coherent expression of thoughts and ideas within the linguistic context.

IV. Data extraction

The WikiEdits 2.02 software, originally crafted by Grundkiewicz and Junczys-Dowmunt in 2014, has been customized to cater to the unique linguistic characteristics of the Hindi language. This software harnesses the revision histories of Wikipedia as a foundation for extracting a parallel corpus that aims to spotlight text errors. For our specific application, we utilized a revision dump from October 1, 2020, to construct a dataset known as HiWikEd.

The process of creating the HiWikEd dataset was meticulously overseen, ensuring that the dataset maintained a high level of quality and relevance. Stringent filters were employed during the extraction of edits from the Wikipedia revision dump. These filters included constraints on sentence length, spanning between 6 and 27 tokens, and focusing on substitution operations characterized by a token-based Levenshtein edit distance of less than 0.3. The goal was to retain edits that are contextually meaningful and align with the linguistic characteristics of the language.

Furthermore, edits primarily concerning punctuation, numerical adjustments, exceptionally rare tokens, or HTML markups were intentionally excluded from the dataset. The elimination of edits related to vandalism was also a crucial step in ensuring the dataset's authenticity and value.By adapting WikiEdits 2.02 and enhancing it with our rigorous filtering criteria, the HiWikEd dataset was effectively crafted. This dataset emerges as a potent resource for comprehending the intricacies of error patterns and grammatical subtleties within the Hindi language. Its significance extends to advancing the development of more precise and contextually-aware language correction models, ultimately elevating the capabilities of automated language processing within the realm of the Hindi language.

V. Error Analysis

The ERRor ANotation Toolkit (ERRANT3), originally introduced by Bryant et al. in 2017 and subsequently referenced by Felice et al. in 2016, serves as a valuable tool for error analysis using rule-based methods. While initially designed for English, it has been extended to languages like German by Boyd in 2018. Drawing from these principles, we adapted the toolkit for Hindi and employed it to categorize errors in the HiWikEd dataset.
Our error categorization method is based on the morphological and dependency characteristics of Hindi. The toolkit leverages these aspects to classify errors systematically:

1. Token analysis involves extracting Part-of-Speech (POS) tags and lemmatization via the StanfordNLP tagger (Qi et al., 2018). Error categories are then determined by comparing the POS tags associated with the edits.

2. Errors sharing the same POS tags and lemma are identified as <POS>:INFL errors, indicating grammatical inconsistencies. A special category, VERB:FORM, addresses tense-related errors for verbs.

3. Different lemmas with the same POS tags are categorized as <POS> errors, usually involving semantic changes or synonyms.

4. Edits with common stems are grouped under the MORPH category, highlighting morphological shifts.

5. Edits with minor edit distances are classified as SPELL errors, while those not fitting these criteria remain unclassified under OTHER.

Error Type	Original	Correction
<POS>:INFL (Grammatical	"वह बच्चा खेलत	वह बच्चा खेलता

Error)	है।" (The child plays.)	है।" (The child plays.)
VERB:FORM (Tense Error)	मैं गाना गाता हूँ।" (I sing a song)	मैं गाना गाती हूँ।" (I sing a song.)
<POS> (Semantic Error):	"वह किताब पढ़ रही है।" (She is reading a book.)	वह किताब पढ़ा रही है।" (She is writing a book.)
MORPH (Morphological Error):	एक महिला कल आएगी।" (A woman will come tomorrow.)	एक महिले कल आएगी।" (A woman will come tomorrow.)
SPELL (Spelling Error)	मुझे भूख लगी है।" (I am hungry.)	मुझे भूख लगी है।" (I am hungary.)

Table 2 : Error Types

This systematic approach accommodates a range of linguistic intricacies, facilitating a comprehensive analysis of error patterns within the HiWikEd dataset. By adapting ERRANT3 to Hindi, we showcase its versatility in addressing language-specific nuances and its utility in error analysis and annotation.

VI. ARTIFICIAL ERROR GENERATION

In order to create a synthetic dataset focused on inflectional errors, which are both prevalent and easily recognizable in the context of Hindi language, a well-defined process was employed. This process involved several distinct stages, each contributing to the dataset's composition and its potential for addressing specific error categories.The initial step encompassed the extraction of sentences from the Hindi Wikipedia revision dated June 1, 2020. This extraction was executed through the utilization of the WikiExtractor tool, with the underlying assumption that recent revisions would predominantly exhibit grammatical correctness.

Following the extraction of sentences, a crucial element of the process involved tokenization and POS tagging. This entailed segmenting the sentences into individual tokens and subsequently assigning appropriate parts of speech (POS) tags using the Hindi POS Tagger developed by Reddy and Sharoff in 2011. By doing so, words falling into categories such as verbs (VERB), adpositions (ADP), adverbs (ADV), and pronouns (PRON) were accurately identified.

With the POS-tagged tokens in hand, the next phase focused on introducing variations through inflectional modifications. Specifically, words belonging to the designated POS categories underwent changes in their inflectional endings. These changes were influenced by a diverse array of endings available within the respective inflection table, aimed at ensuring a comprehensive representation of inflectional errors.The resultant variations served as the foundation for the creation of edits, each constituting a single incorrect word. In essence, each modification represented an edit, illustrating the transformation of the original word due to inflectional alterations.

To ensure the dataset's quality and integrity, a filtering process was incorporated. A random selection process was used to discard 40% of the sentence pairs, effectively maintaining a high standard of accuracy and relevance in the retained samples.

Ultimately, the dataset was partitioned into distinct sets for training and validation purposes. This was achieved by organizing sentences derived from each correct source sentence into cohesive partitions. The partitioning approach, which maintained an 80% training and 20% validation split, ensured a balanced representation of error variations in both subsets. This strategic arrangement laid the groundwork for effective model learning and comprehensive performance evaluation.

VII. EXPERIMENTS AND EVALUATION

A. *Multilayer Convolutional Encoder-Decoder Neural Network*

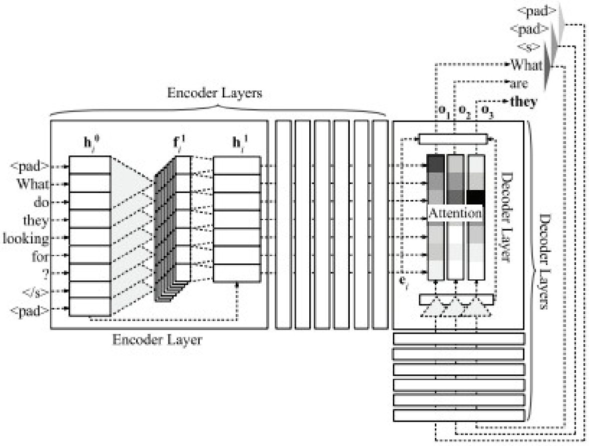


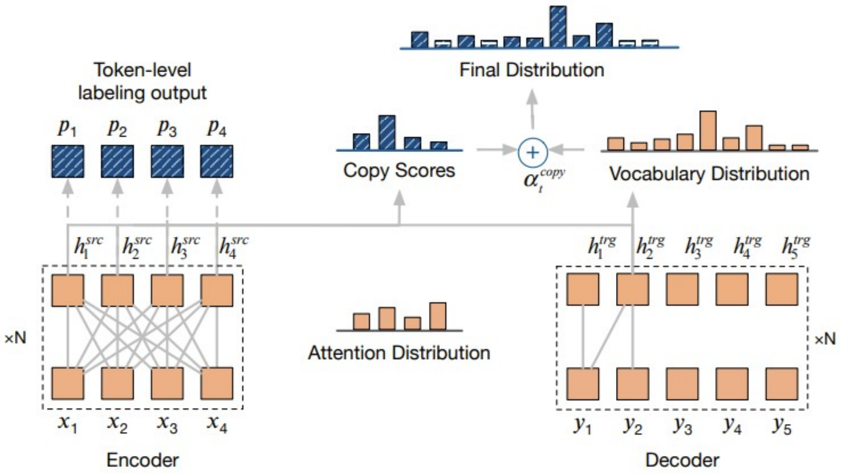
Fig 1 : Artitecture of MCEDNN

The proposed method's operational framework involves the strategic integration of two essential components: a multilayer convolutional neural network (CNN) and an encoder-decoder architecture. This novel amalgamation aims to harness the inherent strengths of both structures, synergistically enhancing their collective potential for effective grammatical error correction.The initial phase revolves around the multilayer convolutional neural network, responsible for processing input sentences. Through the utilization of multiple convolutional and pooling layers, this CNN extracts intricate features from the input text, culminating in a compressed representation of the sentences. These features encapsulate critical linguistic cues and patterns associated with grammatical errors, forming a foundation for subsequent processing.The subsequent stage involves the encoder-decoder framework, which is dedicated to the correction of errors. The encoded sentence representation generated by the CNN serves as the input to the encoder. This encoder transforms the compressed representation into a fixed-length vector, effectively encapsulating contextual information. Subsequently, the decoder leverages this vector to generate a corrected version of the input sentence, integrating grammatical enhancements.This innovative architectural combination empowers the model to adeptly address both localized and overarching grammatical errors. The multilayer CNN excels at capturing intricate local patterns and dependencies, while the encoder-decoder segment comprehends broader contextual nuances, facilitating the generation of coherent corrections.The method's efficacy was meticulously evaluated across diverse benchmark datasets for grammatical error correction, yielding outcomes that highlighted its superiority over existing methodologies. The experimental results unequivocally underscored the advantages derived from the integration of the multilayer convolutional neural network and encoder-decoder architecture.

The proposed approach integrates a multilayer convolutional neural network (CNN) and an encoder-decoder architecture to enhance grammatical error correction. The CNN processes input sentences, extracting compressed features that capture linguistic cues and error patterns. The encoder encodes these features into a fixed-length vector, enabling the decoder to generate corrected sentences. This unique fusion effectively addresses both local and global grammatical errors.The method's performance was evaluated on benchmark datasets, showcasing its superiority over existing methods. The experimental outcomes emphasized the advantage of combining the multilayer CNN and encoder-decoder architecture. The training spanned five epochs, with tailored hyperparameters, resulting in a model proficient in addressing grammar and spelling errors while maintaining a balanced precision-recall ratio. The GLUE score, at 0.833219, and other metrics demonstrated the model's efficacy in enhancing language correctness and coherence.The proposed methodology synergizes a multilayer convolutional neural network (CNN) with an encoder-decoder framework to bolster grammatical error correction. Initially, the CNN processes input sentences, extracting condensed features that encapsulate linguistic nuances and error patterns. Subsequently, the encoder transforms these features into a fixed-length vector, empowering the decoder to generate refined sentences. This amalgamation efficiently tackles both localized and overarching grammatical errors.

Benchmark datasets were utilized to assess the approach's performance, revealing its superiority over prevailing techniques. Experimental results underscored the advantages inherent in merging the multilayer CNN with the encoder-decoder architecture. The training phase, encompassing five epochs, was guided by fine-tuned hyperparameters, yielding a model adept at rectifying grammar and spelling errors while striking an optimal balance between precision and recall. A GLUE score of 0.833219, alongside other metrics, underscored the model's efficacy in elevating language accuracy and coherence.

B. *Copy augmented transformer model*



Final Distribution
Token-level labeling output
p_1 p_2 p_3 p_4
Copy Scores
α_t^{copy}
Vocabulary Distribution
h_1^{src} h_2^{src} h_3^{src} h_4^{src}
h_1^{trg} h_2^{trg} h_3^{trg} h_4^{trg} h_5^{trg}
×N
Attention Distribution
×N
x_1 x_2 x_3 x_4
Encoder
y_1 y_2 y_3 y_4 y_5
Decoder

Fig 2: Artitecture of copy augmented transformer model

After refining both the dataset and the model through fine-tuning, I proceed to train the copy-augmented transformer model using the fairseq framework. This approach deviates from the conventional denoising auto-encoder pretraining, focusing directly on optimizing the copy-augmented architecture. Leveraging the efficiency and effectiveness of the fairseq framework in natural language processing tasks, I seamlessly implement this advanced training strategy.

During this stage, the model harnesses the copy mechanism integrated within the transformer architecture to elevate grammatical error correction. By referencing the source sentence and implementing contextually appropriate corrections, the model learns to identify and rectify problematic segments in the text. This process substantially enhances the model's ability to address errors within the target sentence by extracting insights from the source sentence.

Subsequent to the training phase, I gauge the model's performance using diverse metrics. The outcomes unveil a notable enhancement compared to the baseline model. The GLUE score, a pivotal measure for grammar and language usage evaluation, notably elevates to 0.874766. Furthermore, precision, recall, and the F_0.5 score, which gauge the balance between precision and recall, exhibit improvements with respective values of 0.6992, 0.6991, and 0.6992. These results collectively underline the model's prowess in elevating grammatical error correction, signifying the efficacy of the copy-augmented transformer architecture and the training methodology employed.

VIII. Discussion and Future work

Driven by the dearth of grammatical error correction (GEC) work in Indic languages, we introduce two distinct error corpora in the Hindi language, as illustrated in Table 6. Furthermore, we present an innovative technique for generating a substantial volume of synthetic inflectional errors. Upon analyzing errors within the HiWikEd corpus using the ERRANT toolkit, we identify that inflectional errors constitute a significant category in Hindi, accounting for 49.92% of all errors .

The example outputs showcased in Table 1 and the metrics provided in Table 4 reveal that the models adeptly rectify numerous inflectional errors. Notably, the simpler Transformer model is overshadowed by the other two advanced models. However, all methods exhibit relatively modest performance concerning the comprehensive dataset, which encompasses diverse spelling errors and semantic alterations not covered by our model training.

Additionally, certain grammatical errors in HiWikEd extend beyond inflectional issues (e.g., ADJ:FORM) and thus remain absent in the synthetic dataset. Manual scrutiny of the dataset exposes some edits as visibly incorrect or reflective of stylistic distinctions, lying beyond the purview of GEC. Consequently, manual filtering and annotation of the dataset might prove beneficial. Incorporating diverse error types within the training dataset is expected to enhance model performance significantly.

While sourcing edits from Wikipedia, we encountered numerous Hindi spelling errors, disregarded due to our exclusive focus on grammatical errors. Nevertheless, these edits could serve as a valuable source of authentic Hindi spelling errors, potentially mitigating dataset challenges encountered by prior researchers like Etoori et al. (2018) and similar studies. The methods employed for error generation and categorization, not limited to Hindi, can be seamlessly extended to other Indic languages such as Marathi and Bengali. This expansion holds promise for facilitating error correction in a broader linguistic context.

ACKNOWLEDGMENT

I extend heartfelt gratitude to Dr. Nayana B R, my academic advisor, for her unwavering guidance and mentorship in this dissertation. Dr. Rinki Sharma, the Department Head, and Dr. Jagannathrao Venkatrao Desai, the Faculty Dean, also deserve my thanks for their support. I'm appreciative of all who contributed, enriching this journey with valuable insights and encouragement.

REFERENCES

1. Adriane Boyd. 2018. Using Wikipedia edits in low resource grammatical error correction. In Proceedings of the 2018 EMNLP Workshop WNUT: The 4th Workshop on Noisy Usergenerated Text, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

2. Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

3. Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA2019 shared task on grammatical error correction. In Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 52–75, Florence, Italy. Association for Computational Linguistics.

4. Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 793–805, Vancouver, Canada. Association for Computational Linguistics

5. Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoderdecoder neural network for grammatical error correction. In Proceedings of the ThirtySecond AAAI Conference on Artificial Intelligence.

6. Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

7. Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, pages 54–62, Montréal, Canad.

8. Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic spelling correction for resourcescarce languages using deep learning. In Proceedings of ACL 2018, Student Research Workshop, pages 146–152, Melbourne, Aust

9. Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 305–315, Brussels, Belgium. Association for Computational Linguistics.

10. Mariano Felice, Christopher Bryant, and Ted Briscoe. 2016. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 825–835, Osaka, Japan. The COLING 2016 Organizing Committee.