IndiGEC: Multilingual Grammar Error Correction for Low-Resource Indian Languages

Ujjwal Sharma, Pushpak Bhattacharyya
Computation for Indian Language Technology (CFILT)
Indian Institute of Technology Bombay, Mumbai, India.
(ujjwalsharma, pb)@cse.iitb.ac.in

Abstract

Grammatical Error Correction (GEC) for low-resource Indic languages faces significant challenges due to the scarcity of annotated data. In this work, we introduce the Mask-Translate&Fill (MTF) framework, a novel approach for generating high-quality synthetic data for GEC using only monolingual corpora. MTF leverages a machine translation system and a pretrained masked language model to introduce synthetic errors and tries to mimic errors made by second-language learners. Our experimental results on English, Hindi, Bengali, Marathi, and Tamil demonstrate that MTF consistently outperforms other monolingual synthetic data generation methods and achieves performance comparable to the Translation Language Modeling (TLM)-based approach, which uses a bilingual corpus, in both independent and multilingual settings. Under multilingual training, MTF yields significant improvements across Indic languages, with particularly notable gains in Bengali and Tamil, achieving +1.6 and +3.14 GLEU over the TLM-based method, respectively. To support further research, we also introduce the IndiGEC Corpus, a high-quality, human-written, manually validated GEC dataset for these four Indic languages, comprising over 8,000 sentence pairs with separate development and test splits.

1 Introduction

Grammatical Error Correction (GEC) is a monolingual text-to-text rewriting task where, given a sentence containing grammatical errors, the goal is to produce its grammatically correct version. Modern techniques treat GEC as a translation task, converting ungrammatical text to a correct form. However, these methods rely heavily on supervised data in the form of sentence pairs ("*edits*").

Despite growing interest in GEC, most research has focused on English, primarily due to the lack of benchmark GEC datasets for low-resource languages, especially Indic languages (Sharma and Bhattacharyya, 2025).

In recent years, pre-training on synthetic erroneous data followed by fine-tuning on annotated pairs has become a dominant paradigm, achieving state-of-the-art results in English through various data synthesis techniques (Grundkiewicz and Junczys-Dowmunt, 2019; Lichtarge et al., 2019; Zhao and Wang, 2020; Rothe et al., 2021; Kiyono et al., 2019).

As GEC research expands beyond English, similar synthetic data techniques have been applied to low-resource languages to compensate for the scarcity of annotated data. These methods inject noise into clean text using rule-based, probabilistic, or round-trip translation strategies. While rule-based approaches have shown promising results (Grundkiewicz and Junczys-Dowmunt, 2019; Náplava and Straka, 2019; Sonawane et al., 2020), they require language-specific rules and confusion sets, limiting their scalability and the diversity of generated errors.

Model-based error generation approaches (Xie et al., 2018; Stahlberg and Kumar, 2021) offer improved quality but depend on high-quality seed datasets, typically available only for high-resource languages like English.

This work shifts the focus to low-resource GEC for Indic languages. To address existing limitations, we propose a novel synthetic data generation approach that enables training high-quality GEC systems using only monolingual corpora. Our method leverages two readily available resources:

1. Machine Translation system, and

2. Pretrained Masked Language Model.

The method is generic and can be applied to other low-resource languages as well.

Our contributions are:

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22382–22396

1. **Mask-Translate&Fill (MTF):** We propose Mask-Translate&Fill (MTF), a novel synthetic data generation strategy that relies solely on monolingual corpora for grammatical error correction in low-resource settings. MTF consistently and significantly outperforms other monolingual-based techniques. Despite *not using any bilingual data*, MTF achieves **comparable performance** to the TLM-based method and **significantly outperforms it in Bengali and Tamil** (*+1.6 and +3.14* GLEU, respectively) under multilingual training (Refer Table 3).

2. **The IndiGEC Corpus:** A high-quality, manually validated GEC dataset for four Indic languages: Hindi, Bengali, Marathi, and Tamil. The corpus comprises **2,199** Hindi, **2,801** Bengali, **2,260** Marathi, and **871** Tamil sentence pairs, each with separate development and test splits. The Hindi data is sourced from student writing, whereas the data for the other languages is derived from human-written Wikipedia edits. All sentence pairs are rigorously filtered and manually verified by trained annotators to ensure the presence of genuine grammatical errors and accurate corrections (Refer Section 4).

3. **Multilingual GEC Study:** An empirical analysis demonstrating that multilingual models trained with MTF-generated synthetic data show significant gains in GLEU scores over single-language models across multiple Indic languages. Our experiments report improvements of **+1.91**, **+6.0**, **+3.0**, and **+5.58** GLEU points for Hindi, Bengali, Marathi, and Tamil, respectively. This highlights the strong benefits of cross-lingual transfer for low-resource GEC, achieved without additional data by simply combining monolingual resources (Refer Tables 3 & 4).

The dataset and code are publicly available in the *IndiGEC*[1] repository. To the best of our knowledge, *IndiGEC* is the first publicly available GEC dataset encompassing multiple low-resource Indic languages.

2 Related Works

The primary objective of GEC is to transform an ungrammatical sentence into a grammatical one. Neural Machine Translation (NMT) has emerged as a leading approach for GEC due to its ability to correct errors at the word, phrase, and sentence levels, even when those errors are not seen during training. The transformer architecture (Vaswani et al., 2017) is now the standard for training neural GEC models.

However, a major limitation of transformer-based GEC systems is their reliance on large amounts of supervised data in the form of sentence-level "*edits*" (incorrect-correct pairs). While substantial progress has been made for English and other high-resource languages, supported by numerous benchmark datasets (Ng et al., 2014; Grundkiewicz and Junczys-Dowmunt, 2014; Faruqui et al., 2018; Dale et al., 2012; Bryant et al., 2019), low-resource languages, particularly Indic languages, remain underexplored.

Manually annotated corpora have played a crucial role in advancing GEC for English and other resource-rich languages such as Russian (Rozovskaya and Roth, 2019). However, the creation of such corpora is time-consuming and resource-intensive, and typically infeasible for low-resource languages. To address this, artificial data generation for GEC has gained traction (Izumi et al., 2004; Zhao et al., 2019; Kiyono et al., 2019). Common techniques include injecting noise into clean sentences using rule-based or probabilistic methods, such as token swapping, insertion, or round-trip translation via a pivot language (Lichtarge et al., 2019; Zhao et al., 2019).

Another strategy involves mining corrections from online sources like language learning platforms, Wikipedia revision histories (Faruqui et al., 2018), and GitHub repositories (Hagiwara and Mita, 2020). While this can produce large and natural datasets, many revisions do not address grammatical errors but instead reflect content improvements or rephrasings.

Backtranslation-based methods have also been widely explored to create synthetic datasets (Xie et al., 2018; Stahlberg and Kumar, 2021). These techniques train models on high-quality seed data to generate ungrammatical variants from correct sentences, mimicking real-world errors. However, their effectiveness depends heavily on the availability of such seed datasets.

Finally, masking-based synthetic data generation has also shown promise (Zhao and Wang, 2020; Kaneko et al., 2020). These methods introduce errors by masking parts of clean input sentences and

[1] https://github.com/ujjwalsharmaIITB/IndiGEC

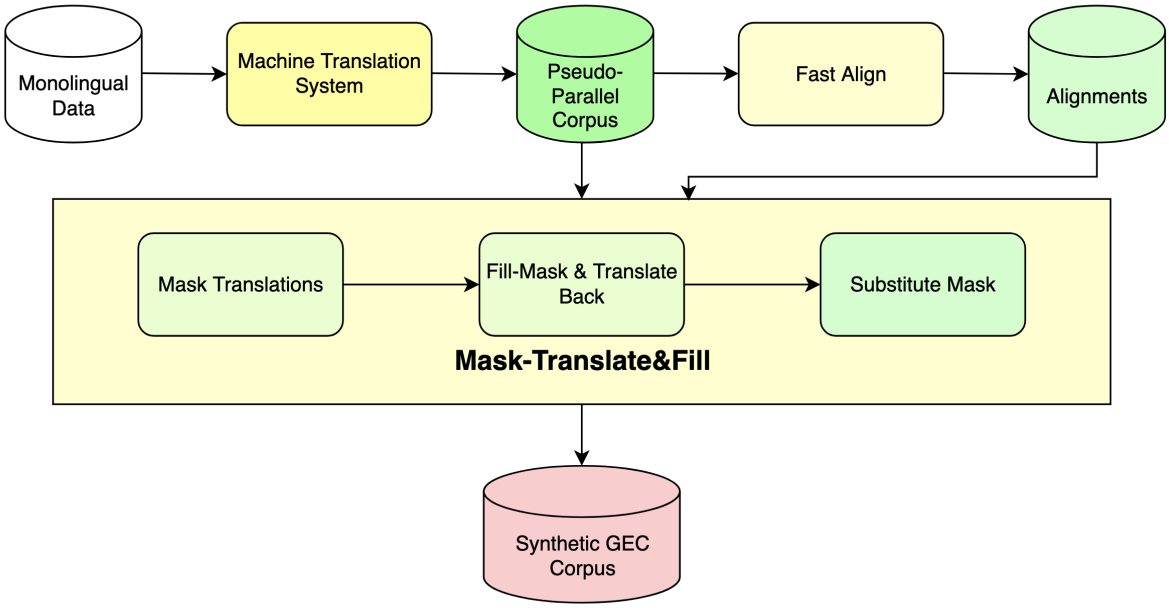

Figure 1: The Mask-Translate&Fill (MTF) pipeline for synthetic GEC data generation. Monolingual sentences are first translated to form a pseudo-parallel corpus. Fast Align is used to compute word alignments, guiding token masking in the target (high-resource) language. The masked sentences are then filled using a language model, and the filled masks back-translated, producing ungrammatical variants that are paired with the original sentences.

replacing them with simulated mistakes. Kaneko et al. (2020) further initialized GEC models using pretrained Masked Language Models (MLMs), followed by fine-tuning.

3 Methodology

In this section, we propose a novel framework called **Mask-Translate&Fill (MTF)** for generating synthetic training data in GEC. MTF integrates translation with masked token prediction to produce more diverse and natural errors. To contextualize our approach, we first review existing masking-based methods, such as *Mask&Fill* and *TLM-based Mask&Fill*, which generate synthetic errors by masking tokens in correct sentences and predicting substitutions. These masking-based approaches are language-agnostic, scalable, and effectively leverage pretrained language models.

3.1 Mask&Fill

In the Mask&Fill approach, we randomly mask *[MASK]* tokens in an input sentence and use a pretrained Masked Language Model (MLM) to predict candidate substitutions. We employ a beam search strategy, masking one token at a time and generating multiple candidate replacements. At each step, the top $beam\_size$ sequences (after substitution) are retained. The final sequence in the beam (i.e., the one with the lowest overall probability after all the substitutions) is selected as the synthetic (ungrammatical) output. The original sentence serves as the corresponding correct target for the GEC model. Formally, let $M = \{m_1, , m_{|M|}\}$ denote the set of masked token positions in the input. The fill-mask process is defined as:

$$S_M = \text{FillMask}_\theta(S, M, [\text{MASK}]) \quad (1)$$

where S is the original sentence, S_M is the generated (potentially erroneous) sentence, and $FillMask_\theta$ denotes the masked language model parameterized by θ.

3.2 TLM based Mask&Fill

This method is based on the approach proposed by Sun et al. (2022). We utilize a pretrained cross-lingual language model (PXLM), trained with the Translation Language Modeling (TLM) objective, to generate synthetic data from a parallel corpus. Following the *Mask&Translate* paradigm, we mask tokens in the target sentence and input the masked sentence into PXLM. Exploiting the model's non-autoregressive generation capability, we sample candidate substitutions for the masked tokens. The sampling procedure is similar to Mask&Translate, and the final sentence in the beam (i.e., the one with the lowest overall probability) is selected as

the ungrammatical input, while the original input sentence is treated as the correct form. For further details, refer to Sun et al. (2022).

3.3 Mask-Translate&Fill

We propose a novel technique, **Mask-Translate&Fill (MTF)**, inspired by the cognitive process of second-language learners. When confronted with a cloze (fill-in-the-blank) task, learners often translate the sentence into their native language, *infer* the missing word, and then translate the completed sentence back into the target language. Our method emulates this behavior as a design heuristic, using it to guide the overall pipeline.[2]

Given an input sentence in a low-resource language (LRL), we first apply token-level masking. The masked sentence is then translated into a high-resource language (HRL), such as English, with the mask preserved. We then use a pretrained MLM in the HRL to fill the masked token. Finally, the filled mask is translated back into the original language and inserted into the original sentence. The masking and sampling process mirrors the Mask&Fill strategy, and the final sentence from the beam (with the lowest probability) is selected as the synthetic (ungrammatical) input. The original sentence serves as the correct target. Figure 1 illustrates this process.

The sampling follows the same formulation as Equation 1, with the key difference that $FillMask_\theta$ now refers to the model in the HRL rather than the LRL.

4 The IndiGEC Corpus

We introduce the **IndiGEC Corpus**, a new human-curated and manually verified GEC corpus for four Indic languages: Hindi, Bengali, Marathi, and Tamil. The corpus includes separate development and test splits for each language, aimed at enabling robust evaluation of multilingual and low-resource GEC systems.

Hindi. For Hindi, data was collected from handwritten notebooks of students from grades 5 to 10. Trained annotators were provided with these notebooks and asked to identify, extract, and correct grammatical errors. Each sentence pair in the dataset comprises a student-written (ungrammatical) sentence and its corresponding corrected version, ensuring high-quality, real-world grammatical error instances.

Bengali, Marathi, and Tamil. For Bengali, Marathi, and Tamil, we extracted sentence-level edits from Wikipedia using the publicly available `wikiedits`[3] tool. Since Wikipedia edits are made by human contributors, they offer a natural source of linguistic corrections. Annotators manually reviewed these edits to identify sentence pairs where the source contained grammatical errors. If the automatically extracted target sentence was inaccurate or stylistically inappropriate, annotators made necessary corrections to ensure the target represented a grammatically sound version of the source.

All sentence pairs across the four languages were manually verified to ensure that they contain genuine grammatical corrections. Table 1 reports the dataset statistics. See Appendix B for further details and analysis.

Language	Dataset	Dev	Test
Hindi	IndiGEC	1100	1099
Bengali	IndiGEC	1400	1401
Marathi	IndiGEC	1130	1130
Tamil	IndiGEC	436	435
English	JFLEG	754 (*4)	747 (*4)
Hindi	Hi-GEC	976	1465

Table 1: **Development and Test Split Overview Across Language Datasets**. This table summarizes the number of sentence pairs in the development (Dev) and test (Test) splits used for evaluation across different GEC datasets. IndiGEC covers Hindi, Bengali, Marathi, and Tamil, while JFLEG and Hi-GEC are English and Hindi benchmarks, respectively. Parentheses indicate multiple reference corrections per sentence.

5 Data

This section outlines the datasets and techniques for synthetic data generation used in our experiments. We conduct experiments across five languages: English, Hindi, Bengali, Marathi, and Tamil. For these experiments, we utilize the ILCI corpus (Jha, 2010), which provides parallel data for English to Hindi, Marathi, Bengali, and Tamil. Before generating synthetic datasets, the corpus was preprocessed to filter sentences with lengths between 10

[2]We emphasize that this analogy is intended purely as an illustrative design heuristic and not as a psychological claim.

[3]https://github.com/snukky/wikiedits

and 100 words. Additionally, we applied regex-based filtering to remove noise and irrelevant content, ensuring the quality of data for subsequent synthetic generation. The statistics for the sentence pairs in the filtered corpus are given in Table 2.

Language Pairs	# Sentences
English-Hindi	135.20k
English-Bengali	94.44k
English-Marathi	85.48k
English-Tamil	82.91k

Table 2: **Total Number of Sentences per Language Pair** in the filtered ILCI Corpus.

5.1 Direct-Noise

Direct-Noise introduces noise into clean sentences using simple operations: word deletion, insertion, replacement, and swapping, each applied with a fixed probability. These transformations support large-scale synthetic data generation, leveraging the abundance of multilingual text online (Izumi et al., 2004; Grundkiewicz and Junczys-Dowmunt, 2019; Lichtarge et al., 2019).

An error rate $p_{errors} \sim \mathcal{N}(0.2, 0.05)$ is sampled per sentence and scaled by sentence length to determine how many tokens to corrupt. Each selected token is modified using one of the following word-level operations, chosen with fixed probabilities: replace (0.3), insert (0.15), delete (0.15), or swap (0.1). The remaining 0.3 is used for character-level noise: skip a character (0.01) or swap adjacent characters (0.06).

5.2 Round-Trip-Translation

Round-trip translation introduces noise by exploiting translation errors and cross-lingual ambiguities. A clean sentence is first translated into a pivot language and then translated back into the original language (Lichtarge et al., 2019).

We use the IndicTrans2 model (Gala et al., 2023) for our experiments. Based on the findings of Sharma and Bhattacharyya (2025), which showed English to be the most effective pivot for Hindi GEC, we adopt English as the pivot for all four Indic languages (Hindi, Bengali, Marathi, and Tamil). For English, we use Hindi as the pivot.

5.3 Mask&Fill

To introduce noise, we sample the error probability as $p_{errors} \sim \mathcal{N}(0.2, 0.05)$. The number of tokens to corrupt in each sentence is computed as $p_{errors} \times$ length(sentence). For English, we use the pretrained BERT model[4] as the $FillMask_\theta$. For Hindi, Bengali, Marathi, and Tamil, we use the IndicBERT model[5]. We adopt a beam search decoding strategy with a $beam\_size$ of 3 in all our experiments.

5.4 TLM-based Mask&Fill

For this method, we use the IndicBERT model[6], pretrained with the TLM objective. We follow the same masking strategy as in the standard Mask&Fill approach, sampling $p_{errors} \sim \mathcal{N}(0.2, 0.05)$ and computing the number of corrupted tokens as $p_{errors} \times$ length(sentence). The $beam\_size$ is set to 3 across all experiments.

5.5 Mask-Translate&Fill

For our proposed MTF method, we employ the IndicTrans2 model (Gala et al., 2023) for translation between English and Indic languages. We use the following pretrained checkpoints:

- `ai4bharat/indictrans2-en-indic-1B` for translation from English to Indic languages
- `ai4bharat/indictrans2-indic-en-1B` for translation from Indic languages to English

As IndicTrans2 does not support translation while preserving masked tokens, we first translate the unmasked sentences and then use `fast_align`[7] to compute word alignments between the source and translated sentences. Using these alignments, we identify and mask the corresponding token in the target sentence to produce aligned masked translations.

We adopt the same noise sampling strategy: $p_{errors} \sim \mathcal{N}(0.2, 0.05)$, with the number of masked tokens computed accordingly. The $beam\_size$ is set to 3 across all experiments.

Note: Masking-based methods, including MTF, predominantly induce substitution-based errors; however, these substitutions can implicitly give rise to *insertion* or *deletion* errors (Appendix D.3).

[4] `google-bert/bert-base-uncased`
[5] `ai4bharat/IndicBERTv2-MLM-Sam-TLM`
[6] `ai4bharat/IndicBERTv2-MLM-Sam-TLM`
[7] https://github.com/clab/fast_align

Source: Translation (English): Gloss:	सभी देशों को आक्रामक रूप से वायरस के लिए तैयार होना चाहिए और सामुदायिक प्रसारण सहित सभी परिदृश्यों के लिए तैयार रहना चाहिए। All countries should be prepared aggressively for the virus and be ready for all scenarios including community transmission. all countries-to aggressive manner-from virus for ready be should and community transmission including all scenarios for ready remain should.
Direct-Noise: Gloss:	सभी देशों को आक्रामक रूप से वायरस के तैयार लिए होना चाहिए और सामुदायिक प्रसारण सहित सभी के लिए तैयार रहना चाहिए। all countries-to aggressive manner-from virus-of ready for be should and community transmission including all for ready remain should.
Round Trip Translation: Gloss:	सभी देशों को वायरस के लिए आक्रामक रूप से तैयार रहना चाहिए और सामुदायिक संचरण सहित सभी परिदृश्यों के लिए तैयार रहना चाहिए। all countries-to virus for aggressive manner-in ready remain should and community transmission including all scenarios for ready remain should.
Mask&Fill: Gloss:	सभी लोगों को आक्रामक रूप से वायरस के लिए तैयार होना चाहिए और सामुदायिक प्रसारण के सभी परिदृश्यों के लिए तैयार रहना चाहिए । all people-to aggressive manner-from virus for ready be should and community transmission-of all scenarios for ready remain should.
Mask-Translate &Fill: Gloss:	सभी देशों को समुदाय रूप से जीका के लिए तैयार होना चाहिए और सामुदायिक प्रसारण सहित सभी संभव है के लिए तैयार रहना चाहिए। all countries-to community manner-from Zika for ready be should and community transmission including all possible is for ready remain should.
TLM based Mask&Fill: Gloss:	सभी देशों को आक्रामक रूप से निपटने के लिए तैयार होना चाहिए और सामुदायिक प्रसारण के संभावित परिदृश्यों के लिए तैयार रहना चाहिए। all countries-to aggressive manner-from dealing for ready be should and community transmission-of possible scenarios for ready remain should.

Figure 2: **Examples of Sentences Generated Using Different Synthetic Data Generation Techniques for GEC in Hindi**. The original sentence (translation and gloss) is provided at the top. Each subsequent row displays an erroneous variant generated by a distinct synthetic data generation technique: Direct-Noise, Round Trip Translation, Mask&Fill, Mask-Translate&Fill (proposed), and TLM-based Mask&Fill. Words that each technique has modified are highlighted in **red**.

6 System Overview

The Error Correction Module (ECM) addresses grammatical errors by framing correction as a machine translation task, translating ungrammatical sentences into grammatical ones using an Encoder-Decoder architecture. The encoder encodes the ungrammatical sentence into a latent representation, and the decoder autoregressively generates the corrected sentence, formalized as:

$$P(y|\hat{y}) = \prod_{i=1}^{n} P(y_i|y_{i-1}, \ldots, y_1, \tilde{y}) \qquad (2)$$

where y is the grammatical output, y_i the word generated at step i, $\hat{y}$ the input sentence, and $\tilde{y}$ its encoded representation. Training maximizes the likelihood of correct sentences, given their corresponding incorrect ones in the dataset D_E, using cross-entropy loss.

6.1 Implementation and Training

All ECMs are implemented using the Transformer architecture (Vaswani et al., 2017). Models are trained exclusively on synthetic datasets and evaluated on held-out test sets. Validation is conducted on the development split of each corresponding language to monitor training progress and prevent overfitting.

We train separate models for five languages: English, Hindi, Bengali, Marathi, and Tamil. For consistency, we use the same amount of synthetic data as outlined in Table 2 for each language, with one sentence pair per sample.

6.1.1 Multilingual Models

To investigate the benefits of cross-lingual transfer for GEC, we train a set of multilingual models covering all five languages. These models vary along two dimensions: (a) **Training Data**: different synthetic data configurations, including multilingual variants; and (b) **Model Size**: varying encoder-

Dataset	English (JFLEG)	Hindi (Hi-GEC)	Hindi	Bengali	Marathi	Tamil
			(IndiGEC Corpus)			
Direct-Noise	33.10	64.02	23.44	*50.3*	36.20	46.72
Round Trip Translation	27.00	40.75	22.50	12.30	16.29	8.12
Mask&Fill	32.27	62.99	20.96	48.17	41.68	44.95
Mask-Translate&Fill (MTF)	*40.23**	*67.95**	*23.30**	47.91	*42.08*	*46.36**
TLM-based Mask&Fill	**40.35***	**70.36***	**23.63***	**52.31***	**45.63***	**48.90***
Multilingual: MTF	***39.83***^{i}	***69.81***^{i}	***22.97***^{i}	***53.91***^{s}	***45.13***^{i}	***52.04***^{s}

Table 3: **Evaluation of Synthetic Data Generation Techniques Across Languages.** GLEU scores on English (JFLEG), Hindi (Hi-GEC), and four Indic languages from the IndiGEC corpus (Hindi, Bengali, Marathi, Tamil) using different synthetic data generation methods: Direct-Noise, Mask&Fill, Mask-Translate&Fill (MTF), and TLM-based Mask&Fill (Sun et al., 2022). The highest score per language is shown in bold; among non-TLM methods, the best score is additionally italicized. "*" denotes statistically significant improvement over Mask&Fill. Superscripts s and i indicate significance and insignificance, respectively, compared to the TLM-based method ($p\_value < 0.05$).

decoder depths while keeping all other parameters constant.

To train the multilingual model, we concatenate the dataset for all five languages and then jointly train a single model.

7 Results

This section presents quantitative and qualitative evaluations of synthetic data generation methods for GEC across English, Hindi, Bengali, Marathi, and Tamil. We compare different data generation techniques, highlighting the effectiveness of our MTF method. We then examine the effects of multilingual training, cross-lingual transfer, and model size on performance. Finally, we qualitatively analyze model outputs to gain insights into error patterns and correction behavior. All evaluations are based on the GLEU metric (Napoles et al., 2015).

7.1 Quantitative Analysis

Table 3 compares the scores across all language pairs. In our experiments, we simulate a low-resource setting for English by restricting access to large-scale annotated data, enabling a fair comparison with genuinely low-resource Indic languages.

MTF consistently outperforms other monolingual synthetic data generation methods. Compared to Mask&Fill, MTF achieves a **+7.96** GLEU gain in English (40.23 vs. 32.27) and **+4.96** in Hindi (Hi-GEC). It also shows improvements on the IndiGEC corpus: for Hindi, MTF attains a **+2.34** increase (23.30 vs. 20.96) and performs comparably to the TLM-based model with only a 0.33 difference (23.63 vs. 23.30). For Tamil, MTF significantly improves by **+1.41** (46.36 vs. 44.95) over Mask&Fill. For Marathi and Bengali, it maintains comparable performance (42.08 vs. 41.68 and 47.91 vs. 48.17, respectively). These statistically significant improvements (highlighted in **bold** where applicable) demonstrate MTF's strong effectiveness despite relying solely on monolingual data.

Under multilingual training, the model achieves significant improvements over its single-language MTF variant. Specifically, Hindi (Hi-GEC) improves by **+1.91**, while the IndiGEC corpus shows gains of **+6.0** for Bengali, **+3.0** for Marathi, and **+5.58** for Tamil, with performance remaining comparable in English. Notably, these improvements are achieved without any additional data, simply by combining the same amount of monolingual synthetic data across languages, underscoring the effectiveness of cross-lingual transfer for low-resource GEC. Furthermore, under multilingual training, MTF achieves *performance comparable to* the TLM-based method across most languages. Notably, for Bengali and Tamil, the multilingual MTF model ***significantly outperforms*** the TLM-based approach, as confirmed by statistical significance testing.

Table 4 illustrates the impact of increasing data in multilingual training. Using only the lowest-probability sequence in MTF yields higher or comparable GLEU scores across most languages compared to using all the outputs in the final beam (*All Beam*). This suggests that selecting the lowest-probability sequence provides cleaner, more effective training data, while including all beam outputs increases data volume but may introduce noise that slightly reduces performance.

Table 5 shows that increasing model size does

Dataset	English (JFLEG)	Hindi (Hi-GEC)	Hindi	Bengali	Marathi	Tamil
			(IndiGEC Corpus)			
Mask-Translate&Fill (MTF)	39.83	69.81	22.97	53.91	45.13	52.04
MTF: All Beam ($beam\_size = 3$)	39.90	67.58	23.59	52.25	43.28	50.39

Table 4: **Performance of Multilingual Models Trained with Mask-Translate&Fill Synthetic Data.** GLEU scores for English (JFLEG), Hindi (Hi-GEC), and four Indic languages from the IndiGEC corpus (Hindi, Bengali, Marathi, Tamil). The *All Beam* variant uses all outputs from the final beam ($beam\_size = 3$) instead of selecting only the lowest-probability sequence, thereby increasing the amount of synthetic training data.

Model	English (JFLEG)	Hindi (Hi-GEC)	Hindi	Bengali	Marathi	Tamil
			(IndiGEC Corpus)			
4 Encoder - 4 Decoder	39.83	69.81	22.97	53.91	45.13	52.04
6 Encoder - 6 Decoder	39.82	67.79	23.26	52.09	43.43	50.26
8 Encoder - 8 Decoder	39.97	69.41	22.67	53.87	44.30	51.68

Table 5: **Effect of Encoder-Decoder Size on Multilingual GEC Performance.** GLEU scores for multilingual models with varying encoder-decoder sizes (4E-4D, 6E-6D, 8E-8D) across English (JFLEG), Hindi (Hi-GEC), and four Indic languages (Hindi, Bengali, Marathi, Tamil) from the IndiGEC corpus. The models were trained using the Mask-Translate&Fill synthetic data.

not lead to consistent performance gains. The 4E-4D model achieves the best or comparable results across most languages, including Hindi, Bengali, Marathi, and Tamil. Larger models (6E-6D and 8E-8D) often perform slightly worse, suggesting overfitting in low-resource settings. This indicates that *model size is not the bottleneck* in our case, possibly because the MTF data generation strategy is effective, enabling efficient learning even with smaller models. For detailed category-wise analysis, see Appendix C.

Setup	Mask&Fill	MTF
Dev: IndiGEC Test: Hi-GEC	63.26	**68.01**
Dev: Hi-GEC Test: IndiGEC	21.26	**22.96**
Dev: Hi-GEC Test: Hi-GEC	62.99	**67.95**
Dev: IndiGEC Test: IndiGEC	20.96	**23.30**

Table 6: Performance comparison of Mask&Fill and MTF in a cross-domain setting using *Hindi* datasets differing in register and writing style.

7.1.1 Impact of Data Source Heterogeneity

To assess the impact of style/register differences, we conducted a cross-domain evaluation using two Hindi datasets: Hi-GEC (Wikipedia-based) and IndiGEC-Hindi (student writing). Table 6 reports GLEU scores for *Mask&Fill* and *MTF* under different domain setups. Results from Table 6 show that *MTF* consistently outperforms the Mask&Fill across domains, indicating stronger robustness and better generalization across registers and writing styles.

VERB Error

Correct: Translation: Gloss:	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर की थी। I had ordered some books from Ka Kha Ga Publications. Ka Kha Ga Publications from some books order done had.
Erroneous: Gloss:	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर कि थी। Ka Kha Ga Publications from some books order that had.
Direct-Noise	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर कि थी।
RTT:	क ख गजा प्रकाशन से कुछ पुस्तकें ही थी ।
Mask&Fill:	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर कि थी ।
MTF:	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर की थी।
TLM based:	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर कि थी।
Multilingual (MTF):	क ख ग प्रकाशन से कुछ पुस्तकें ऑर्डर की थी।

Figure 3: Comparison of correct and erroneous Hindi sentences illustrating a verb form error where the conjunction "*ki*" is incorrectly substituted for the feminine past participle verb "*kī*". The table includes translations, glosses, and outputs from various error correction models.

7.2 Qualitative Analysis

We examine model outputs to gain deeper insight into the effectiveness of different synthetic data generation methods: Direct-Noise, RTT, Mask&Fill, MTF, TLM-based, and the multilingual MTF model. This qualitative evaluation focuses on how well each model identifies and corrects

common grammatical errors, providing an intuitive understanding of their strengths and limitations.

Figure 3 illustrates a common Hindi verb inflection error where the conjunction *"ki"* is incorrectly used instead of the feminine past participle verb *"kī"*. This example demonstrates that only the MTF and Multilingual MTF models successfully detect and correct this subtle morphological mistake, reflecting their deeper grasp of language-specific grammatical nuances learned from the MTF-generated data. For detailed qualitative analysis, see Appendix D.

8 Conclusion and Future Work

In this work, we propose Mask-Translate&Fill (MTF), a synthetic data generation strategy for grammatical error correction that relies solely on monolingual corpora. Our experiments across English and four Indic languages: Hindi, Bengali, Marathi, and Tamil, show that MTF outperforms other monolingual synthetic data methods, including Direct-Noise, Round-Trip-Translation, and Mask&Fill. Multilingual training with MTF-generated data yields significant improvements over single-language models, performs comparably to the TLM-based method, and significantly outperforms it for Bengali and Tamil, demonstrating the benefits of cross-lingual transfer in low-resource GEC using only monolingual data. Our analysis indicates that model size is not a major performance bottleneck, likely due to the effectiveness of MTF-generated data in enabling learning with smaller models. Qualitative evaluations further show that MTF captures subtle, language-specific grammatical patterns, particularly in verb morphology and proper noun usage, and introduces a richer variety of syntactic errors during training. In addition, we release the IndiGEC Corpus, a high-quality, manually validated dataset for four Indic languages, to facilitate benchmarking and further research. Future work could focus on expanding the diversity of errors generated, especially those found in advanced or native-level writing, including the incorporation of semantic and pragmatic errors to broaden the scope and applicability of synthetic data.

All data and code are publicly available[8] under the CC BY-SA 4.0 license.

[8] https://github.com/ujjwalsharmaIITB/IndiGEC

Limitations

While the Mask-Translate&Fill (MTF) synthetic data technique is particularly effective in low-resource settings, its performance is influenced by the availability of pretrained translation and masked language models. In cases where such models are limited or unavailable, especially for extremely low-resource languages, the quality of generated data may vary. Additionally, MTF primarily simulates a focused class of second-language learner errors through a fill-in-the-blank mechanism. While this captures common grammatical patterns effectively, it may not encompass the full spectrum of errors found in more advanced or native-level writing, offering an opportunity for future work to expand the diversity of error types.

Ethics Statement

Our work uses multiple data sources for training and evaluation, including the ILCI corpus for training, and a combination of Wikipedia edits, student writing samples, and public benchmarks for evaluation. For Wikipedia-based datasets, we acknowledge the ethical considerations of using crowd-sourced content. All extracted data was anonymized and aggregated to ensure that no individual contributors are identifiable. For student writing data, we obtained appropriate permissions from schools and guardians, and annotations were conducted under ethical guidelines. Annotators, including those with expertise in the respective languages, maintained strict confidentiality and were fairly compensated for their work. For the JFLEG benchmark, we used publicly released data in accordance with its terms of use. We recognize that all data sources may carry inherent biases, particularly crowd-sourced content like Wikipedia, and these may affect model behavior. We encourage continued efforts to improve fairness, representation, and transparency in low-resource GEC systems.

References

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 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.

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, Canada. Association for Computational Linguistics.

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.

Jay Gala, Pranjal A. Chitale, Raghavan AK, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar, Janki Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M. Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Preprint*, arXiv:2305.16307.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *Advances in Natural Language Processing – Lecture Notes in Computer Science*, volume 8686, pages 478–490. Springer.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-augmented grammatical error correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.

Masato Hagiwara and Masato Mita. 2020. GitHub typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6761–6768, Marseille, France. European Language Resources Association.

Emi Izumi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. The overview of the SST speech corpus of Japanese learner English and evaluation through the experiment on automatic detection of learners' errors. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Girish Nath Jha. 2010. The TDIL program and the Indian langauge corpora intitiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 177–184, Boston, MA. Association for Machine Translation in the Americas.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The

case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Ujjwal Sharma and Pushpak Bhattacharyya. 2025. HiGEC: Hindi grammar error correction in low resource scenario. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6063–6075, Abu Dhabi, UAE. Association for Computational Linguistics.

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.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

Xin Sun, Tao Ge, Shuming Ma, Jingjing Li, Furu Wei, and Houfeng Wang. 2022. A unified strategy for multilingual grammatical error correction with pre-trained cross-lingual language model. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4367–4374. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628, New Orleans, Louisiana. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1226–1233.

A Implementation Details

We utilized the *OpenNMT*[9] library (Klein et al., 2018) for training all our models.

During training, each model is validated on the validation split of the datasets and evaluated on an unseen test set. Early stopping is implemented with patience of 5 epochs, and models are validated at the end of each epoch.

Table 7 provides a comprehensive overview of the hyperparameters used for training the models

Hyperparameter	Value
Encoder Layers	4
Decoder Layers	4
Hidden Size	512
Word-Vector Size	512
Feed-Forward Size	1024
Activation @ Feed Forward	gelu
Multi-Head Attention Heads	16
Optimizer	Adam
Initial Learning Rate	1.0
Early Stopping Patience	5
Attention Dropout	0.1

Table 7: Hyperparameters used for training the models. Only the number of layers changed from 4 to 6 to 8 in multilingual experiments.

Data tokenization is performed using sentencepiece tokenizer with the `spm-library`[10], with a vocabulary size of 32,000.

B The IndiGEC Corpus

This section presents an analysis of The *IndiGEC* Corpus, introduced in Section 4. The corpus was manually constructed by extracting and validating human-written texts in four Indian languages: Hindi, Bengali, Marathi, and Tamil.

B.1 Hindi

For Hindi, we collected notebooks of students from grades 5 to 10 from various schools. Trained annotators were tasked with identifying, extracting, and correcting erroneous sentences from these notebooks.

B.2 Bengali, Marathi, and Tamil

For Bengali, Marathi, and Tamil, we extracted sentence-level edits from Wikipedia using the pub-

[9] https://opennmt.net/

[10] https://github.com/google/sentencepiece

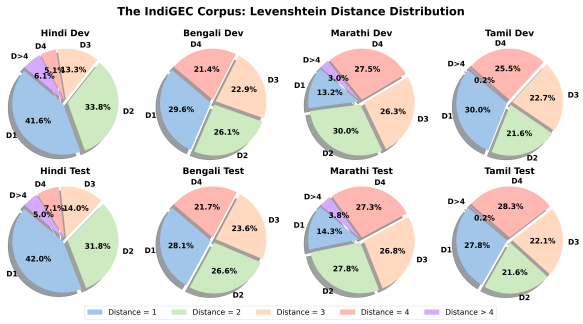

Figure 4: Distribution of Levenshtein distances between sentence pairs in the dev and test sets for each language in the IndiGEC corpus.

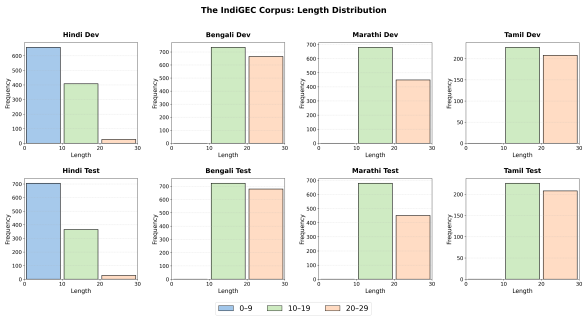

Figure 5: Length distribution of target sentences in the dev and test sets across the four Indic languages in the IndiGEC corpus.

licly available `wikiedits`[11] tool. These Wikipedia edits, contributed by human editors, serve as a natural source of linguistic corrections.

B.2.1 Wikipedia Edit Histories

Wikipedia maintains comprehensive revision histories for all pages, providing snapshots of page content before and after every edit. Each pair of consecutive revisions represents a single change.

To extract edits, we used the `wikiedits` tool with significant enhancements. We added improved filtering and cleaning processes, such as normalization and tokenization using the `IndicNLP Library`[12], and applied the tool to Wikipedia revision dumps dated April 1, 2025, for Bengali[13], Marathi[14], and Tamil[15].

We filtered the edits using the following constraints:

- Sentence length was restricted to between 6 and 26 words.
- Edits were retained only if they involved no more than 4 word-level changes and had a Levenshtein edit ratio below 0.35.
- Following Sharma and Bhattacharyya (2025), we discarded edits that involved only punctuation or numeric changes, rare tokens, HTML markup, vandalism, or identical sentence pairs.

All extracted edits were manually reviewed by annotators to ensure quality. They verified whether the source sentence contained grammatical errors and confirmed that the corresponding target sentence was a fluent and grammatically correct revision. If the automatically extracted target sentence was inaccurate or stylistically awkward, annotators revised it accordingly.

All annotations and corrections were carried out by trained annotators holding master's degrees in the respective languages, some of whom are high-school teachers, ensuring both linguistic expertise and consistency across the corpus. All annotators were fairly compensated for their work.

Figure 4 presents pie charts illustrating the distribution of Levenshtein distances between sentence pairs in the dev and test sets for each language. Most edits fall within a distance of 1 to 4, with very few exceeding 4, indicating minimal but meaningful corrections.

Figure 5 displays histograms showing the length distribution of target sentences in the dev and test sets across all four languages.

C Quantitative Analysis

To compare the strengths of the Multilingual MTF model against the TLM-based approach, we analyzed GLEU scores across specific error categories. Figure 6 presents the category-wise GLEU scores for Mask&Fill, MTF, TLM, and Multilingual MTF on the Hi-GEC test set. While both the Multilingual MTF and TLM-based models show strong overall performance, Multilingual MTF surpasses TLM in several key error types, demonstrating its ability to capture complex grammatical phenomena despite relying solely on monolingual data. In particular, Multilingual MTF achieves higher scores in **ADJ:INFL** (adjective inflection), indicating im-

[11] https://github.com/snukky/wikiedits

[12] https://github.com/anoopkunchukuttan/indic_nlp_library

[13] https://dumps.wikimedia.org/bnwiki/20250401/bnwiki-20250401-pages-meta-history.xml.7z

[14] https://dumps.wikimedia.org/mrwiki/20250401/mrwiki-20250401-pages-meta-history.xml.7z

[15] https://dumps.wikimedia.org/tawiki/20250401/tawiki-20250401-pages-meta-history.xml.7z

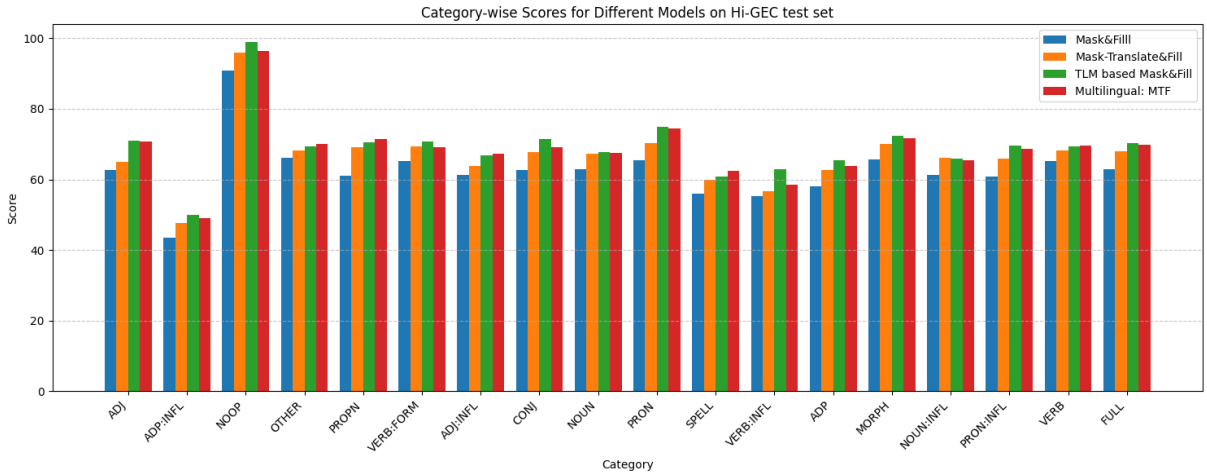

Figure 6: **Category-wise GLEU scores on the Hi-GEC test set** for different models: Mask&Fill, MTF, TLM-based, and Multilingual MTF. Multilingual MTF achieves higher scores than the TLM-based model in several key categories, including adjective inflection (ADJ:INFL), proper nouns (PROPN), spelling (SPELL), verbs (VERB), and complex/uncategorized edits (OTHER), despite relying only on monolingual data.

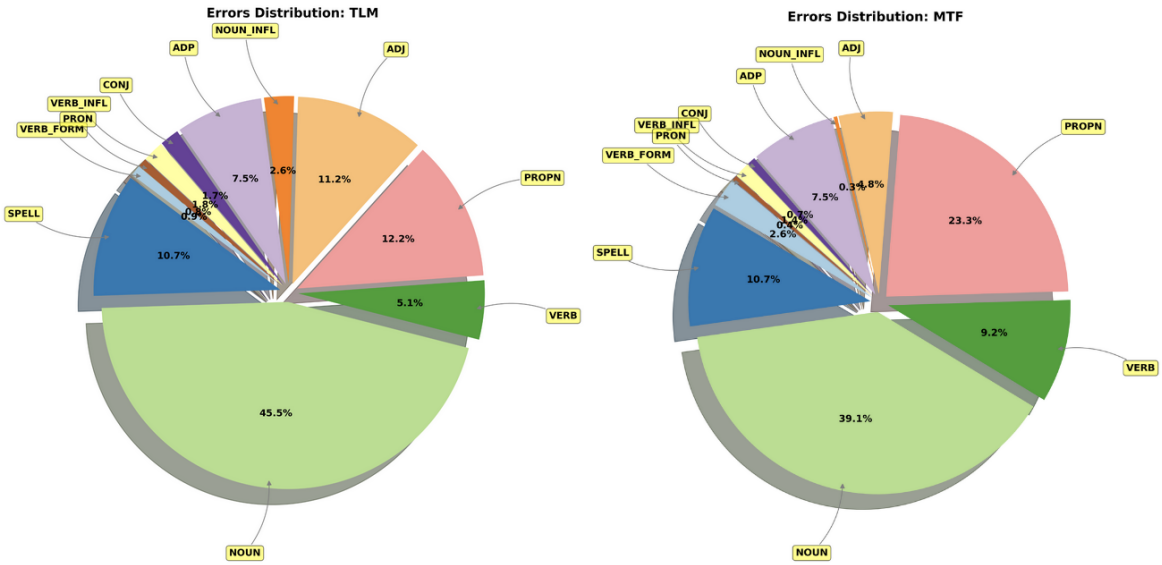

Figure 7: Distribution of Error Types in Hindi Synthetic Data Generated by TLM and MTF.

proved handling of morphological agreement in gender, number, and case, common challenges in Indic languages. It also performs better in **PROPN**, reflecting stronger modeling of proper noun usage and context-sensitive case marking. In **SPELL**, MTF benefits from the masked language model used during data generation, resulting in more accurate spelling corrections. Its gains in **VERB** suggest better modeling of verb morphology and syntactic structure. Additionally, in the **OTHER** category, representing more complex or uncategorized edits, Multilingual MTF demonstrates better generalization, likely due to the greater diversity of its synthetic training data. Importantly, unlike the TLM-based model, which relies on parallel corpora and a pretrained cross-lingual model trained using the TLM objective, MTF achieves comparable or superior performance using only monolingual corpora. This highlights MTF's potential as a scalable and resource-efficient solution for grammatical error correction in low-resource settings.

D Qualitative Analysis

D.1 Synthetic Data Distribution

To better understand model behavior on specific error types, we randomly sampled 1,000 Hindi sentences each from the MTF and TLM-generated datasets and analyzed them using the Hindi extension of the ERRANT toolkit[16]. Figure 7 presents the distribution of common error types across the two datasets.

The analysis reveals distinct differences in error type distributions between MTF and TLM-generated Hindi data. MTF produces a higher proportion of verb-related errors, approximately 4.0% in total (2.6% verb form + 1.4% verb inflection), compared to 2.7% for TLM (0.9% verb form + 1.8% verb inflection). Additionally, MTF generates more proper noun errors (23.3% vs. 12.2%) and slightly more main verb errors (9.2% vs. 5.1%).

Hi-GEC - VERB

Correct: Translation: Gloss:	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाती है। Mermaids are found in many stories and legends. Mermaids many stories and legends in found are.
Erroneous: Gloss:	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाति है। Mermaids many stories and folk tales in caste are.
Mask&Fill:	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाति है।
MTF:	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाती है।
TLM based:	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाति है।
Multilingual (MTF):	जलपरियां कईं कहानियों व दंत कथाओं में पाई जाती है।

Hi-GEC - PRON

Correct: Translation: Gloss:	यह मंदिर एक झील के बीचोबीच बना हुआ है । This temple is built in the middle of a lake. This temple a lake in middle built is.
Erroneous: Gloss:	ये मंदिर एक झील के बीचोबीच बना हुआ है । these temple one lake GEN middle built be.
Mask&Fill:	ये मंदिर एक झील के बीचोबीच बना हुआ है ।
MTF:	ये मंदिर एक झील के बीचोबीच बना हुआ है ।
TLM based:	यह मंदिर एक झील के बीचोबीच बना हुआ है ।
Multilingual (MTF):	ये मंदिर एक झील के बीचोबीच बना हुआ है ।

Figure 8: Examples of common grammatical errors in Hindi (verb inflection and pronoun agreement) and corresponding model outputs. The top row shows a verb error corrected only by the MTF-trained models, while the bottom row shows a pronoun error corrected only by the TLM-trained model. These examples illustrate the distinct strengths of models trained on different synthetic data.

Conversely, TLM yields a significantly higher proportion of adjective errors (11.2% vs. 4.8%) and noun errors (45.5% vs. 39.1%), along with more pronoun and conjunction errors (0.8% vs. 0.4%, and 1.7% vs. 0.7%, respectively). The proportion of spelling and adposition errors remains similar across both methods.

D.2 Model Outputs

Figure 8 presents examples from Hindi that illustrate two common error types: (i) a verb inflection error, where *jāti* (caste) is incorrectly used instead of *jātee* (goes, feminine singular), and (ii) a pronoun agreement error, where *ye* (plural) is used in place of *yah* (singular).

In the first example (verb inflection error), only the model trained on MTF data correctly identifies and corrects the error; the models trained on Mask and TLM data fail to modify the input. In the second example (pronoun error), the TLM-trained model produces the correct correction, while the Mask and MTF-based models again leave the input unchanged.

These examples highlight how MTF and TLM each capture different error patterns, suggesting that the choice of data generation method can significantly impact a model's ability to generalize to specific linguistic phenomena.

Marathi-Misspelling

Correct: Translation: Gloss:	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते बौद्ध धर्मातील क्षत्रिय परंपरेत या दिवसाचे औचित्य आहे . In Buddhism, the day of Akshaya Tritiya is considered especially important; in the Kshatriya tradition of Buddhism, this day holds significance. Buddhist religion-in Akshaya this day special importance considered goes Buddhist religion Kshatriya tradition-in this day significance is.
Erroneous: Gloss:	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते बौद्धध धर्मातील क्षत्रिय परंपरेत या दिवसाचे औचित्य आहे . Buddhist religion-in Akshaya this day special importance considered goes Buddhadh religion Kshatriya tradition-in this day significance is.
Mask&Fill:	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते बौद्धध धर्मातील क्षत्रिय परंपरेत ह्या दिवसाचे औचित्य आहे .
MTF:	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते आणिध धर्मातील क्षत्रिय परंपरेत या दिवसाचे औचित्य आहे .
TLM based:	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते बौद्धध धर्मातील क्षत्रिय परंपरेत या दिवसाचे औचित्य आहे .
Multilingual (MTF):	बौद्ध धर्मामध्ये आखाजी या दिवसाचे विशेष महत्व मानले जाते , धर्मातील क्षत्रिय परंपरेत या दिवसाचे औचित्य आहे .

Figure 9: Example of a spelling error in Marathi and the corresponding model outputs. The original typo ("bauddhadh") is retained by the TLM-based and Mask&Fill models, while MTF and Multilingual MTF detect the error and modify the erroneous segment through substitution and deletion, respectively.

Figure 9 illustrates a typographical misspelling error in Marathi, where the word "bauddha" ("Buddhist") is incorrectly written as "bauddhadh" with an extra "dh" appended. This misspelling does not form any valid word in Marathi and introduces noise in the sentence.

Among the outputs generated, both the TLM-based model and the Mask&Fill model retained the typo in their outputs. The Mask&Fill model

[16] https://github.com/s-ankur/errant

Language	% Sentences with Token Difference	Average Token Difference
Hindi	57.00%	2.12
Bengali	35.80%	1.45
Marathi	19.80%	1.26
Tamil	16.05%	1.18
Average	32.16%	1.50

Table 8: Statistics of token-level differences across languages. The percentage indicates the proportion of sentences containing at least one token change (insertion or deletion), while the average token difference measures the mean number of insertions and deletions per sentence.

attempted to mitigate the anomaly by appending "hyā" ("this") after the erroneous phrase, but it failed to correct the original misspelling. The TLM-based model did not react to the error at all, simply reproducing the typo unchanged, indicating a lack of sensitivity to such noise. In contrast, the MTF and Multilingual (MTF) models showed a stronger ability to detect the anomaly. The MTF model tried to correct the corrupted phrase but replaced it with another malformed word, "āṇidh", which is also invalid in Marathi, suggesting partial recognition but poor substitution. The Multilingual MTF model took a different route by completely removing the corrupted word "bauddhadh", retaining only "dharmatīl" ("of the religion"). While this helped eliminate the typo, it also resulted in a partial loss of meaning.

Overall, while none of the models produced a fully accurate correction, MTF and Multilingual MTF demonstrated a degree of error awareness, marking a positive step toward handling typographical noise, even if their recovery strategies were imperfect.

D.3 Token-Level Error Analysis

All masking-based methods, including the proposed **MTF**, can implicitly cause *insertion* and *deletion* of tokens, even though they primarily perform substitutions. This occurs because replacing a single token with multiple tokens effectively inserts new tokens, while replacing it with no token effectively deletes it, thereby altering the overall token count. To assess this phenomenon, we conducted a token difference analysis on MTF-generated data across four languages: Hindi, Bengali, Marathi, and Tamil. This analysis measures the proportion of sentences whose token counts differ from their original counterparts, as well as the average magnitude of these differences. The results, presented in Table 8, show the percentage of sentences containing any token difference and the average number of token differences per sentence for each language.

As shown in Table 8, the proportion of sentences exhibiting token differences varies considerably across languages. **Hindi** shows the highest impact, with 57% of sentences containing token count changes and an average difference of 2.12 tokens, indicating frequent and relatively larger edits. **Bengali** shows a moderate level of change (35.8%, 1.45 tokens on average), while **Marathi** (19.8%, 1.26) and **Tamil** (16.05%, 1.18) exhibit fewer and smaller token differences. On average, 32.66% of sentences across these languages contain token count changes, with an average magnitude of 1.50 tokens per affected sentence. These results confirm that although MTF primarily performs substitutions, it can implicitly introduce insertions and deletions, especially in morphologically richer or syntactically flexible languages like Hindi and Bengali.