Corpora Generation for Urdu Grammatical Error Correction

Burhanuddin Aliasghar Ezzi
Department of Computer Science
Habib University
Karachi, Pakistan
be07724@st.habib.edu.pk

Ahmed Ali Aun
Department of Computer Science
Habib University
Karachi, Pakistan
aa07600@st.habib.edu.pk

Syed Ahad Ali
Department of Computer Science
Habib University
Karachi, Pakistan
sa07753@st.habib.edu.pk

***Abstract*—Grammatical Error Correction (GEC) for Urdu, a widely spoken South Asian language, remains an under-researched area due to the lack of annotated datasets. This paper addresses the challenge of generating a robust corpus for training deep learning models aimed at Urdu GEC. We propose a method for synthesizing a large dataset by leveraging naturally occurring errors from the Urdu WikiEdits dataset and inflicting similar errors onto a large corpus of correct sentences. Our pipeline aligns sentence pairs, identifies grammatical errors, and inflicts these errors stochastically to produce correct and incorrect sentence pairs. We employ several low-resource techniques, adapting approaches used for other languages like Hindi, to create a high-quality dataset.**

I. Introduction

URDU is a language with a rich heritage and a script influenced by Persian, Arabic, Hindi, Sanskrit, and Turkish. It is widely spoken in South Asian countries, including Pakistan and India, by a large population. Despite its complexity and the variety of grammatical components, unfortunately, there has been little work done in the field of Urdu grammatical error correction (GEC). An automated GEC system for Urdu would greatly enhance content creation, digital communication tools, and educational platforms. To build a state-of-the-art model that generalizes well to a wide range of human grammatical errors and outputs correct sentences for grammatically incorrect inputs, we need a large corpus of training data.

Although there isn't a vast corpus available for this task, there are proven methods (used for other languages) that can help generate such datasets. Our research mainly aims to investigate these techniques and apply them to the Urdu language to create a robust corpus of Urdu.

Our strategy to create a large corpus for Urdu GEC first involves extraction of correct and incorrect sentence pairs from the wiki edits dataset and use it to learn different types of errors in that dataset. Learning errors from a natural source such as Wikiedits will allow us to build a large number of error-infliction rules that can be generalized to the set of all words in the Urdu grammar. This strategy of mimicking a human source can help us focus only on errors that occur in human texts and build a state-of-the-art model. We used a large corpus of precompiled Urdu sentences to inflict the learned errors and generated a new corpus of synthetic error-generated pairs. We will then test different models that are suitable for GEC tasks and compare them for performance. A detailed description of this is given in the Dataset section.

II. Research Question

In our research, we are building a deep neural network that can be used to correct grammatical errors in Urdu. Currently, there is no GEC model for Urdu; therefore, we first focus on building a model that can output correct sentences for input sentences that are grammatically incorrect but require a single edit (insertion, deletion, or substitution) of words for correction and can also detect spelling mistakes and correct them. The task of grammatical error correction is highly complex and requires a lot of considerations; therefore, the model needs to learn properties of grammar such as sentence structure, the set of correct word instances, voice, gender, tense, etc. Therefore, to build such a model, we require a large corpus of data to train the model on. There is no vast corpus of datasets to train our model, but fortunately, there are ways that have proven to be useful (for other languages) for the generation of such a corpus, given there is a small corpus of well-generalized grammatical errors.

In our research, we are tackling two primary research questions: (1) "Generation of synthetic datasets for a low-resource language like Urdu." and (2) "Building a state-of-the-art model that generalizes well to the set of humanly possible grammatical errors". In our research, we have explored different strategies to generate such a corpus that helps us solve our first research question. To build a state-of-the-art model, we will use newly built deep learning models that are better suited for such a task and compare the performance across different models and datasets.

The motivation for this research stems from the lack of tools, such as Grammarly for English, that automatically correct grammatical errors in written Urdu and help users communicate more effectively. An automated GEC for Urdu would positively impact content creation, digital communication tools, and learning platforms. In the case of Indo-Aryan languages such as Urdu, progress in the field of GEC has been limited due to the unavailability of annotated datasets. This gap motivates us to generate a synthetic dataset for Urdu to address this need.

III. Literature Review

There is a notable lack of research on Grammar Error Correction for the Urdu language. The reason is that Urdu is a low-resource language with not many annotated datasets available. To solve this we focused our research mainly on how to generate good quality synthetic data by generating errors on a corpus of the language. For this, we looked to other low-resource languages and the research done on them for synthetic data generation.

Sonawane et al.generate a parallel corpus of synthetic errors by inserting errors into grammatically correct sentences using a rule-based process, focusing specifically on inflectional errors [1]. They generate this artificial dataset for training of models and for evaluating those models, they scrape the edit history of Wikipedia and use that as a real-world data source to test the trained model on. This paper encourages that their strategies on Hindi can be applied to other Indic fusional languages such as Urdu, Bengali, etc.

Lichtarge et al. provides a more general case approach to generate large corpora [2]. The first strategy for corpora generation uses the Wiki edits dataset, which benefits from a more accurate distribution of natural grammatical errors made by humans. However, Wikipedia source contains more text on highly popular topics, and therefore the result can get skewed in favor of those pages. To mitigate this issue, Litcharge et al. discard pages that exceed 64 megabytes and down-sample consecutive revisions from individual pages, selecting only $log_{1.5}(n)$ pairs for a page with a total of n revisions. This reduces the total amount of data 20-fold. Texts are aligned (using a min-edit rule (unsure)), and cuts are introduced at random positions and source-target pairs are generated. Spelling errors are introduced probabilistically in the source sequences at a rate of 0.003 per character, randomly selecting deletion, insertion, replacement, or transposition of adjacent characters for each introduced error. The second approach they use for data generation the is of round-trip machine translation through a high-resource bridge language like English, French, German, Japanese and Russian. They further augment this data with the common errors found in the Wiki edit approach and introduce them here using stochastic methods. This research also finds that using incremental edits produces a significant improvement in performance over single-shot decoding for models trained on the Wikipedia revision data, a highly noisy corpus, while models trained on the relatively clean round-trip translation data see no improvement. The authors pre-train the model on the above Wikipedia revision dataset (both methods) and then fine-tune the model on a lang-8 dataset which is a parallel corpus for GEC but with high noise, Wikipedia-derived data are demonstrated to benefit significantly from finetuning on Lang-8(Tables 6 and 7) because Wikipedia revision dataset contains corrections that are outside the scope of GEC. Finetuning on noisy (the target may still contain errors), helps the model to only make conservative edits that fall under the scope of GEC. The paper also compares different permutations of training sets and finds that the model trained on all types of synthetically generated data gives the best generalization.

Gomez et al. explore three methods for generating synthetic data [3]. Those are random character and token transformations, confusion sets from a spell checker, and round-trip translation as described by Lichtarge et al. [2]. Their approach involves first fine-tuning on synthetic datasets, followed by fine-tuning on gold-standard datasets, with a smaller Seq2Seq model showing strong performance when pre-trained on synthetic data. For data generation, they employed round-trip translation using English as an intermediary, focusing on introducing errors via confusion sets rather than using entire back-translated sentences. The synthetic data was derived from the Ukrainian partition of Common Crawl (CC-100), and transformations included spell-based (using aspell for highly confusable words) and POS-based substitutions. These techniques effectively simulated the target errors for their tasks.

Foster et al. paper introduces GenERRate, a probabilistic error-infliction tool that generates errors such as POS insertions, deletions, and substitutions [4]. It performs well in some scenarios but has limitations, particularly in handling covert errors like sentence structure improvements, which particularly are not problematic with Litcharge et al [2]. The tool's error generation is based on a manual analysis of 89 common errors, primarily involving changes in noun/verb number or article deletion, applied to a large corpus (BNC). This synthetic data shows improved accuracy and recall, especially useful for small-to-medium learner corpora. However, GenERRate struggles with corpora that contain spelling errors, highlighting the need to model such mistakes in error generation. Combining natural and synthetic data helps mitigate performance degradation, showing promise for augmenting training sets with artificial data.

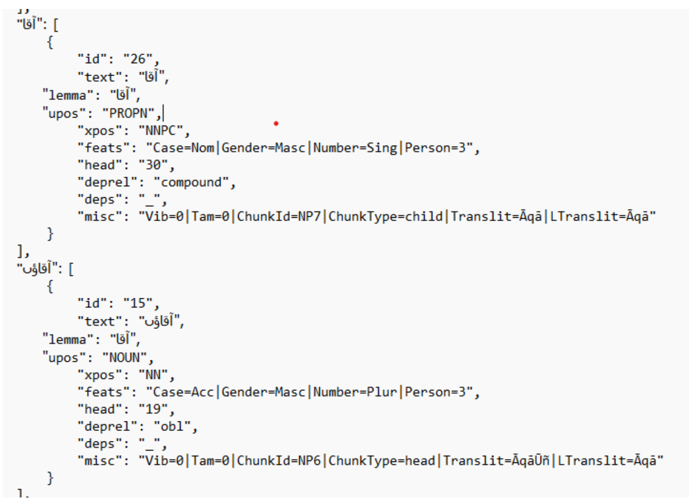
IV. Dataset

The primary focus of our research is Corpora Generation; the generation of dataset, pairs of correct and incorrect sentences, on which we will train our model is the biggest challenge in our research. The only natural source of dataset that can be directly used for our research is the dataset of Urdu wikiedits. The wikiedits dataset comprises of natural errors. This dataset will be use to learn the types of errors which can then be used on other datasets such as scrapped dataset from websites for error infliction. Below is the description of the datasets used:

A. Datasets that are used in the Corpora Generation Pipeline

1) Universal Dependency Treebank for Urdu: We found a Universal Dependency Treebank for Urdu. We used the training data from this dataset which comprises of 4000+ sentences that experts have annotated for morphological and POS features [5] [6]. In Universal Dependencies (UD) for Urdu, tokenization generally follows whitespace to delimit words, with punctuation treated as separate tokens. Morphology in Urdu employs all 17 universal POS categories, with specific attention to particles (PART) and auxiliary verbs (AUX).

This dataset is converted into a word dictionary that provides an efficient way for word substitution during corpora generation.



```
"آقا": [
    {
        "id": "26",
        "text": "آقا",
    "lemma": "آقا",
    "upos": "PROPN",|
        "xpos": "NNPC",
        "feats": "Case=Nom|Gender=Masc|Number=Sing|Person=3",
        "head": "30",
        "deprel": "compound",
        "deps": "_",
        "misc": "Vib=0|Tam=0|ChunkId=NP7|ChunkType=child|Translit=Āqā|LTranslit=Āqā"
    }
],
"آقاؤں": [
    {
        "id": "15",
        "text": "آقاؤں",
    "lemma": "آقا",
    "upos": "NOUN",
        "xpos": "NN",
        "feats": "Case=Acc|Gender=Masc|Number=Plur|Person=3",
        "head": "19",
        "deprel": "obl",
        "deps": "_",
        "misc": "Vib=0|Tam=0|ChunkId=NP6|ChunkType=head|Translit=ĀqāŪñ|LTranslit=Āqā"
    }
],
```

Figure 1: Example of the dictionary generated from the UDTB dataset

2) 150000 Urdu words Dataset: This dataset has been taken from an online dataset from the Urduhack GitHub repository [7]. This dataset will help us to check for spelling mistakes and pre-processing in our corpora generation pipeline.

B. Corpora Generation Pipeline

The Corpora generation will be done using the datasets discussed above. The corpora Generation consists of the following steps:

3) WikiEdits Dataset for Urdu: This dataset will work as a natural source of grammatical errors for Urdu [9]. We will use this dataset to learn the types of errors and use it alongside the 7000-word dictionary to inflict errors on the dataset below.

4) 500 thousand Urdu sentences Dataset: This dataset will be used for the corpora generation. The error types learned with the wiki edits dataset will be used to inflict errors in this dataset stochastically [8].

Lemma	**Words**
کتاب	کتابوں, کتاب, کتابیں
والا	والے, والا, والوں, والو
کمرا	کمرے, کمروں, کمرا
میں	میں, اس, ان, یہ, وہ, ہم, تم, مجھ, تجھ
بیٹھنا	بیٹھنے, بیٹھے, بیٹھا, بیٹھ, بیٹھی, بیٹھتے, بیٹھتا, بیٹھیں, بیٹھنا, بیٹھوں, بیٹھتی, بیٹھتیں, بیٹھو, بیٹھیے, بیٹھایا, بیٹھاتے
کم	کے, کی, کم, کمتر
لئے	لئے
فرش	فرش
علاوہ	علاوہ
کوئی	کوئی
کرسی	کرسی, کرسیاں, کرسیوں

Table I: A subset of words from the lemma dictionary that is made from these sentences using lemma prediction model from UrduHack library

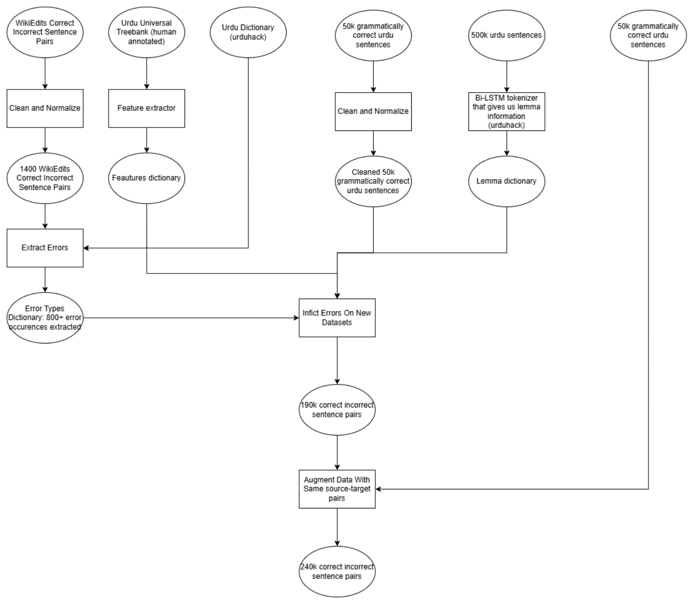
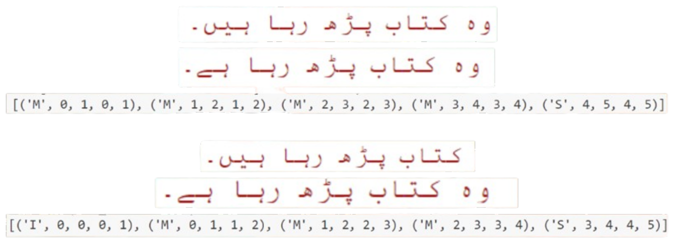

WikiEdits Correct Incorrect Sentence Pairs
Urdu Universal Treebank (human annotated)
Urdu Dictionary (urduhack)
50k grammatically correct urdu sentences
500k urdu sentences
50k grammatically correct urdu sentences
Clean and Normalize
Feature extractor
Clean and Normalize
Bi-LSTM tokenizer that gives us lemma information (urduhack)
1400 WikiEdits Correct Incorrect Sentence Pairs
Feautures dictionary
Cleaned 50k grammatically correct urdu sentences
Lemma dictionary
Extract Errors
Error Types Dictionary: 800+ error occurences extracted
Infict Errors On New Datasets
190k correct incorrect sentence pairs
Augment Data With Same source-target pairs
240k correct incorrect sentence pairs

Figure 2: High Level diagram of the corpora generation pipeline. The highest level represents the external datasets used. The circles represent datasets while squares represent a function

1) Preprocessing: The pre-processing includes constructing the 7000-word dictionary and the extraction of correct and incorrect sentences pairs from the wiki edits dataset.

2) Alignment of sentence pairs and erroneous word identification: : In this process, we align the sentence pairs in the previous step to identify the type of single-word error (substitution, insertion, or deletion). An example is shown below. We took this algorithm from the ERRANT Github repository and modified the code for support in Urdu using the Urduhack library [10] [11].



وہ کتاب پڑھ رہا ہیں۔

وہ کتاب پڑھ رہا ہے۔

```
[('M', 0, 1, 0, 1), ('M', 1, 2, 1, 2), ('M', 2, 3, 2, 3), ('M', 3, 4, 3, 4), ('S', 4, 5, 4, 5)]
```

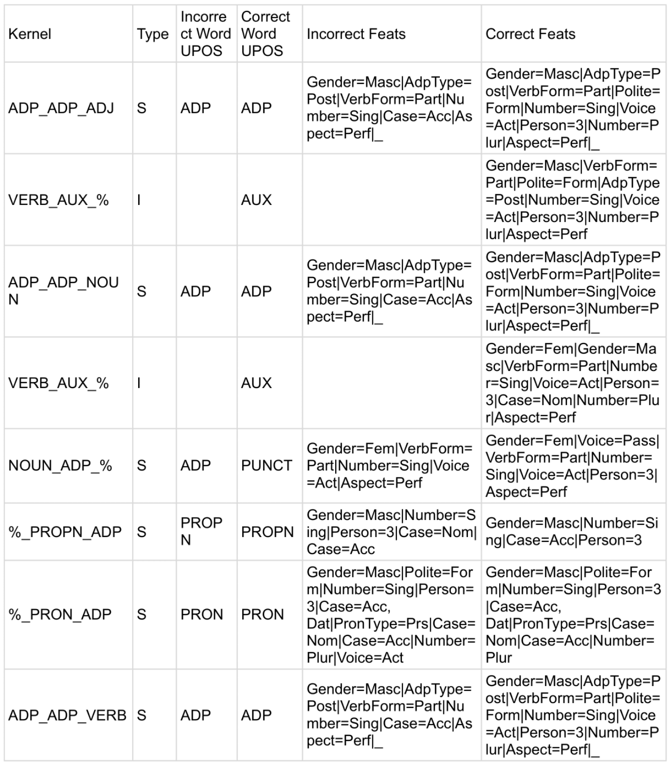
کتاب پڑھ رہا ہیں۔

وہ کتاب پڑھ رہا ہے۔

```
[('I', 0, 0, 0, 1), ('M', 0, 1, 1, 2), ('M', 1, 2, 2, 3), ('M', 2, 3, 3, 4), ('S', 3, 4, 4, 5)]
```

Figure 3: There are two example outputs of the alignment algorithm. The first is the incorrect sentence while the second is the corrected version. The output of the first example shows that the sentence has 5 words in total, 4 matching words "M", and 1 substituted word "S". The output of the second example shows that there has been an insertion "I" of the word at the first index and a substitution at the last index.

3) Error Type Feature Extraction: We extract the types of grammatical errors from the cleaned Wikedits corpus of 1400 sentences and store them. The alignment done above helps to get the type of edit, but after that, we extract two important features of the edit: the structure of the edit, which is represented by the kernel, and the transition of word POS and features. There are three types of errors: substitution (represented by an S), insertion (represented by an I), and deletion (represented by a D). The Kernel is extracted from the POS of the word that on which (or near) the error occurs. The POS is predicted using a model from the UrduHack library. The features of the incorrect and correct words are also incorporated into the type information. There are a total of 860 error occurrences found in the wikiedits dataset which can be classified into 348 types based on their structure, POS, and features.



Kernel	Type	Incorrect Word UPOS	Correct Word UPOS	Incorrect Feats	Correct Feats
ADP_ADP_ADJ	S	ADP	ADP	Gender=Masc|AdpType=Post|VerbForm=Part|Number=Sing|Case=Acc|Aspect=Perf|_	Gender=Masc|AdpType=Post|VerbForm=Part|Polite=Form|Number=Sing|Voice=Act|Person=3|Number=Plur|Aspect=Perf|_
VERB_AUX_%	I		AUX		Gender=Masc|VerbForm=Part|Polite=Form|AdpType=Post|Number=Sing|Voice=Act|Person=3|Number=Plur|Aspect=Perf
ADP_ADP_NOUN	S	ADP	ADP	Gender=Masc|AdpType=Post|VerbForm=Part|Number=Sing|Case=Acc|Aspect=Perf|_	Gender=Masc|AdpType=Post|VerbForm=Part|Polite=Form|Number=Sing|Voice=Act|Person=3|Number=Plur|Aspect=Perf|_
VERB_AUX_%	I		AUX		Gender=Fem|Gender=Masc|VerbForm=Part|Number=Sing|Voice=Act|Person=3|Case=Nom|Number=Plur|Aspect=Perf
NOUN_ADP_%	S	ADP	PUNCT	Gender=Fem|VerbForm=Part|Number=Sing|Voice=Act|Aspect=Perf	Gender=Fem|Voice=Pass|VerbForm=Part|Number=Sing|Voice=Act|Person=3|Aspect=Perf
%_PROPN_ADP	S	PROPN	PROPN	Gender=Masc|Number=Sing|Person=3|Case=Nom|Case=Acc	Gender=Masc|Number=Sing|Case=Acc|Person=3
%_PRON_ADP	S	PRON	PRON	Gender=Masc|Polite=Form|Number=Sing|Person=3|Case=Acc, Dat|PronType=Prs|Case=Nom|Case=Acc|Number=Plur|Voice=Act	Gender=Masc|Polite=Form|Number=Sing|Person=3|Case=Acc, Dat|PronType=Prs|Case=Nom|Case=Acc|Number=Plur
ADP_ADP_VERB	S	ADP	ADP	Gender=Masc|AdpType=Post|VerbForm=Part|Number=Sing|Case=Acc|Aspect=Perf|_	Gender=Masc|AdpType=Post|VerbForm=Part|Polite=Form|Number=Sing|Voice=Act|Person=3|Number=Plur|Aspect=Perf|_

Figure 4: 10 out of the 348 error types extracted from the wikiedits corpus.

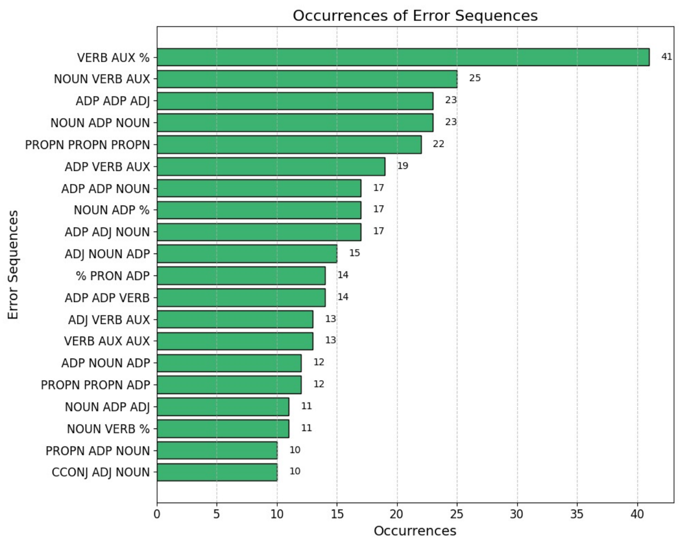
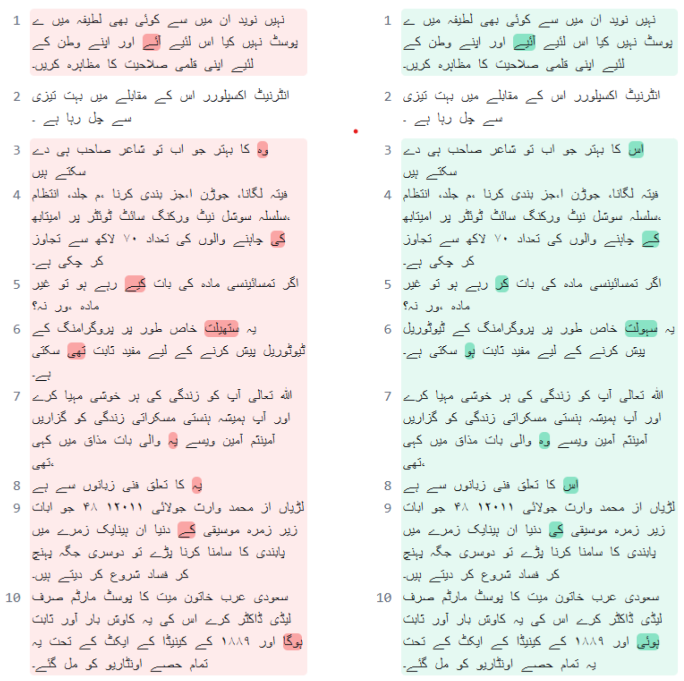


Figure 5: Occurence share of each type of error grouped by their kernel (structure).

4) Error Infliction: The infliction process is done on the cleaned dataset of 100k cleaned sentences. 50k of them are used to generate incorrect sentences while the other 50k sentences are augmented after the incorrect sentence pairs are generated. The generation process picks each sentence from the dataset and generates an incorrect sequence and type of the error that is used. The generator checks if any type of error can be successfully generated on that sentence. It is possible that the generator generate no pair or more than one pair for a given clean sentence based on the possibility. Our generated generates approximately 190k correct incorrect source and targets after which the 50k correct correct source targets are augmented, resulting in a 240k final dataset size that is used for training the model



1 نہیں نوید ان میں سے کوئی بھی لطیفہ میں ے یوسٹ نہیں کیا اس لئیے آئے اور اپنے وطن کے لئیے اپنی قلمی صلاحیت کا مظاہرہ کریں۔
2 انٹرنیٹ اکسپلورر اس کے مقابلے میں بہت تیزی سے چل رہا ہے .
3 وہ کا بہتر جو اب تو شاعر صاحب ہی دے سکتے ہیں
4 فیتہ لگانا، جوڑن اءجز بندی کرنا ،ہم جلد، انتظام ،سلسلہ سوشل نیٹ ورکنگ سائٹ ٹوئٹر پر امیتابھ کی چاہنے والوں کی تعداد ۷۰ لاکھ سے تجاوز کر چکی ہے۔
5 اگر تمسائینسی مادہ کی بات کیے رہے ہو تو غیر مادہ ،ور نہ؟
6 یہ ستھیلت خاص طور پر پروگرامنگ کے ٹیوٹوریل پیش کرنے کے لیے مفید ثابت تھی سکتی ہے۔
7 اللہ تعالی آپ کو زندگی کی ہر خوشی مہیا کرے اور آپ ہمیشہ ہنستی مسکراتی زندگی کو گزاریں آمینثم آمین ویسے یہ والی بات مذاق میں کہی تھی۔
8 یہ کا تعلق فنی زبانوں سے ہے
9 لڑیاں از محمد وارث جولائی ۱۲۰۱۱ ۴۸ جو ابات زیر زمرہ موسیقی کے دنیا ان ہیناپک زمرے میں پابندی کا سامنا کرنا پڑے تو دوسری جگہ پہنچ کر فساد شروع کر دیتے ہیں۔
10 سعودی عرب خاتون میت کا پوسٹ مارٹم صرف لیڈی ڈاکٹر کرے اس کی یہ کاوش بار آور ثابت ہوگا اور ۱۸۸۹ کے کینیڈا کے ایکٹ کے تحت یہ تمام حصے اونٹاریو کو مل گئے۔

1 نہیں نوید ان میں سے کوئی بھی لطیفہ میں ے یوسٹ نہیں کیا اس لئیے آئیے اور اپنے وطن کے لئیے اپنی قلمی صلاحیت کا مظاہرہ کریں۔
2 انٹرنیٹ اکسپلورر اس کے مقابلے میں بہت تیزی سے چل رہا ہے .
3 اس کا بہتر جو اب تو شاعر صاحب ہی دے سکتے ہیں
4 فیتہ لگانا، جوڑن اءجز بندی کرنا ،ہم جلد، انتظام ،سلسلہ سوشل نیٹ ورکنگ سائٹ ٹوئٹر پر امیتابھ کے چاہنے والوں کی تعداد ۷۰ لاکھ سے تجاوز کر چکی ہے۔
5 اگر تمسائینسی مادہ کی بات کر رہے ہو تو غیر مادہ ،ور نہ؟
6 یہ سہولت خاص طور پر پروگرامنگ کے ٹیوٹوریل پیش کرنے کے لیے مفید ثابت ہو سکتی ہے۔
7 اللہ تعالی آپ کو زندگی کی ہر خوشی مہیا کرے اور آپ ہمیشہ ہنستی مسکراتی زندگی کو گزاریں آمینثم آمین ویسے وہ والی بات مذاق میں کہی تھی۔
8 اس کا تعلق فنی زبانوں سے ہے
9 لڑیاں از محمد وارث جولائی ۱۲۰۱۱ ۴۸ جو ابات زیر زمرہ موسیقی کی دنیا ان ہیناپک زمرے میں پابندی کا سامنا کرنا پڑے تو دوسری جگہ پہنچ کر فساد شروع کر دیتے ہیں۔
10 سعودی عرب خاتون میت کا پوسٹ مارٹم صرف لیڈی ڈاکٹر کرے اس کی یہ کاوش بار آور ثابت ہوئی اور ۱۸۸۹ کے کینیڈا کے ایکٹ کے تحت یہ تمام حصے اونٹاریو کو مل گئے۔

Figure 6: Sample snapshot of the generated dataset. The errors are inflicted stochastically. The output from the pipeline is also augmented with the clean dataset for better model generalization

V. Methodology & Results

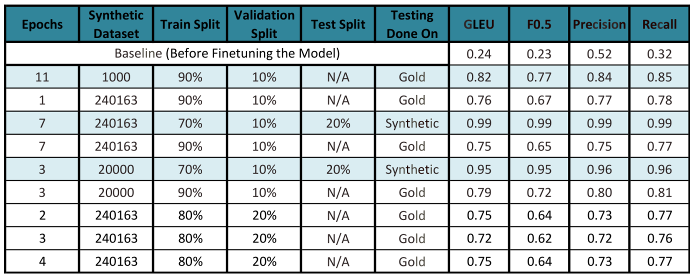
The primary focus of our research was to generate a high-quality synthetic dataset to train a model for Urdu Grammar Error Correction. After generating the synthetic dataset, we obtained 240,000 pairs of sentences in which 190,000 were incorrect-correct pairs and 50,000 were correct-correct pairs. Additionally, we used a gold dataset comprising 1,200 human-written sentence pairs, specifically reserved for testing our fine-tuned model.

We selected the mT0 transformer model for our work, a fine-tuned version of mT5 specifically designed to follow human instructions across dozens of languages in a zero-shot setting. Built upon mT5—a multilingual variant of Google's T5 (Text-to-Text Transfer Transformer). The mT0 model extends the architecture to support multiple languages, making it particularly well-suited for cross-lingual tasks. By fine-tuning these pre-trained multilingual language models on a diverse cross-lingual task mixture (xP3), mT0 achieves robust generalization across languages and tasks, enabling it to effectively handle unseen scenarios [12].

Our hyperparameters for the training loop are as shown in Table II

Hyperparameter	**Value/Description**
Optimizer	AdamW
Learning Rate	3e-4
Weight Decay	1e-5
Learning Rate Scheduler	StepLR
Step Size (Scheduler)	10
Decay Factor (Gamma)	0.5
Batch Size	4
Gradient Accumulation	Gradients accumulated over 4 batches
Gradient Clipping	Maximum gradient norm of 1.0
Mixed-Precision Training	GradScaler for dynamic loss scaling
Text Input Processing	Padding sequences to max length of 128

Table II: Training Loop Hyperparameters



Epochs	Synthetic Dataset	Train Split	Validation Split	Test Split	Testing Done On	GLEU	F0.5	Precision	Recall
Baseline (Before Finetuning the Model)						0.24	0.23	0.52	0.32
11	1000	90%	10%	N/A	Gold	0.82	0.77	0.84	0.85
1	240163	90%	10%	N/A	Gold	0.76	0.67	0.77	0.78
7	240163	70%	10%	20%	Synthetic	0.99	0.99	0.99	0.99
7	240163	90%	10%	N/A	Gold	0.75	0.65	0.75	0.77
3	20000	70%	10%	20%	Synthetic	0.95	0.95	0.96	0.96
3	20000	90%	10%	N/A	Gold	0.79	0.72	0.80	0.81
2	240163	80%	20%	N/A	Gold	0.75	0.64	0.73	0.77
3	240163	80%	20%	N/A	Gold	0.72	0.62	0.72	0.76
4	240163	80%	20%	N/A	Gold	0.75	0.64	0.73	0.77

Figure 7: Performance comparison of fine-tuning the model with varying configurations

Figure 7 shows different parameter values we trained our model on and obtained the relevant scores. These parameters were number of epochs, different sizes of datasets, the train-test split of the data, and the data upon which the testing was done (either on the 1,200 human-written sentence pairs or on small portion of the training data itself. As visible in the figure, our observation was that the our model was giving better scores on three epochs as compared to larger number of epochs, we tried to confirm this by taking a dataset size of 2000 sentences and 30 epochs and our hypothesis was confirmed, as shown in Figure 8

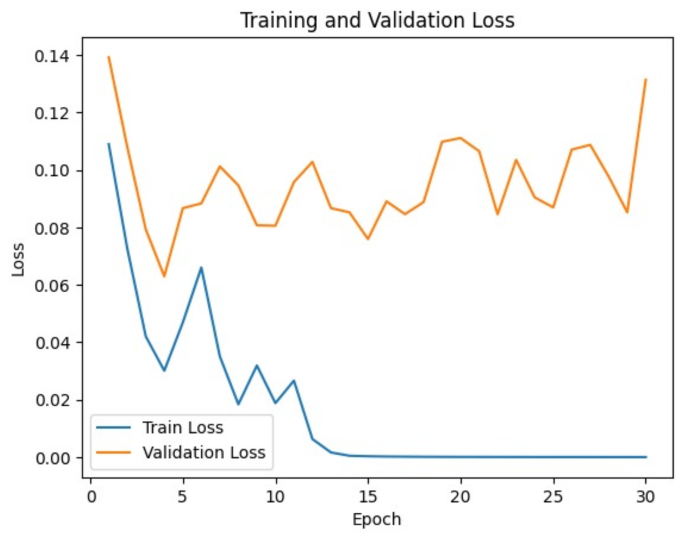


Figure 8: Loss graph of 2000 training dataset size on 30 epochs

We infer that our training can be better assessed on more researched testing datasets.

VI. Limitations & Future Work

While our research contributes significantly to the development of an Urdu Grammatical Error Correction (GEC) system, there are certain limitations and opportunities for further exploration:

A. Limitations

- **Synthetic Data Bias:** Although the synthetic dataset closely mimics human errors, it might not capture all nuances and error patterns seen in real-world text.
- **Limited Testing on Gold Data:** Due to the scarcity of gold-standard datasets for Urdu, the evaluation primarily relied on synthetic datasets, which could limit generalization.
- **Computational Constraints:** Training models such as mT0 for extended epochs and larger datasets required significant computational resources, which constrained experimentation with more complex architectures.
- **Focus on Single-Edit Errors:** The current pipeline primarily targets single-edit errors, leaving multi-edit and more complex errors underexplored.
- **Lack of Diverse Contexts:** The dataset may lack diverse contexts and domains, potentially limiting the model's adaptability to various text types.

B. Future Work

- **Expansion of Gold Data:** Collecting and annotating a more extensive gold-standard dataset will help evaluate models under real-world conditions and improve fine-tuning.
- **Handling Complex Errors:** Extending the pipeline to include multi-edit and covert error types, such as sentence restructuring and contextual errors, will enhance robustness.
- **Domain Adaptation:** Fine-tuning the model for specific domains, such as educational material or formal communication, could improve performance in specialized applications.
- **Incorporation of Larger Models:** Leveraging more powerful models like GPT-4 or similar, adapted for Urdu, might provide better results with nuanced understanding.
- **Community Collaboration:** Engaging with the Urdu-speaking academic and linguistic communities can help build richer datasets and refine error infliction techniques.
- **Cross-Lingual Transfer Learning:** Exploring transfer learning from high-resource languages like Hindi, which shares linguistic similarities with Urdu, can potentially enhance performance.
- **Evaluation on Real-World Use Cases:** Testing the model in practical applications, such as spell-checkers and writing aids, will provide insights into its effectiveness and areas of improvement.

References

[1] A. Sonawane, S. K. Vishwakarma, B. Srivastava, and A. Kumar Singh, "Generating Inflectional Errors for Grammatical Error Correction in Hindi," ACLWeb, Dec. 01, 2020. https://aclanthology.org/2020.aacl-srw.24/.

[2] J. Lichtarge, C. Alberti, S. Kumar, Noam Shazeer, N. Parmar, and S. Tong, "Corpora Generation for Grammatical Error Correction," Jan. 2019, doi: https://doi.org/10.18653/v1/n19-1333.

[3] F. P. Gomez, A. Rozovskaya, and D. Roth, "A Low-Resource Approach to the Grammatical Error Correction of Ukrainian," Jan. 2023, doi: https://doi.org/10.18653/v1/2023.unlp-1.14.

[4] J. Foster and Ø. E. Andersen, "GenERRate: Generating Errors for Use in Grammatical Error Detection," pp. 82–90, Jan. 2009, doi: https://doi.org/10.3115/1609843.1609855.

[5] Bhat, R. A., Bhatt, R., Farudi, A., Klassen, P., Narasimhan, B., Palmer, M., … Others. (n.d.). The Hindi/Urdu Treebank Project. In Handbook of Linguistic Annotation. Springer Press.

[6] Palmer, M., Bhatt, R., Narasimhan, B., Rambow, O., Sharma, D., & Xia, F. (2009). Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In The 7th International Conference on Natural Language Processing (pp. 14–17).

[7] Urduhack. (n.d.). GitHub - urduhack/urdu-words: A text file containing 150,000 Urdu words for all your dictionary/word-based projects e.g: auto-completion / autosuggestion. GitHub. https://github.com/urduhack/urdu-words

[8] Jawaid, B., Kamran, A., & Bojar, O. (2014, May). A Tagged Corpus and a Tagger for Urdu. In LREC (Vol. 2, pp. 2938-2943).

[9] urwiki dump progress on 20240601. (n.d.). https://dumps.wikimedia.org/urwiki/20240601/

[10] Chrisjbryant. (n.d.). GitHub - chrisjbryant/errant: ERRor ANnotation Toolkit: Automatically extract and classify grammatical errors in parallel original and corrected sentences. GitHub. https://github.com/chrisjbryant/errant

[11] Chrisjbryant. (n.d.). GitHub - chrisjbryant/errant: ERRor ANnotation Toolkit: Automatically extract and classify grammatical errors in parallel original and corrected sentences. GitHub. https://github.com/chrisjbryant/errant

[12] BigScience, ”mT0: Multilingual T0 Model,” Hugging Face https://huggingface.co/bigscience/mt0-base