Detection and Correction of Grammatical Errors

Rakinul Haque[1*], Nowreen Tarannum[2†], Naimur Rahman[3†], Hamim Ahmad[4†], Kazi Nazibul Islam[5†], Annajiat Alim Rasel[6†], Sadiul Arefin[7†], Sania Azhmeel[8†]

[1*]Computer Science and Engineering, BRAC University.
[2]Computer Science and Engineering, BRAC University.
[3]Computer Science and Engineering, BRAC University.
[4]Computer Science and Engineering, BRAC University.
[5]Computer Science and Engineering, BRAC University.
[6]Computer Science and Engineering, BRAC University.
[7]Computer Science and Engineering, BRAC University.
[8]Computer Science and Engineering, BRAC University.

*Corresponding author(s). E-mail(s): rakinulhaque3@gmail.com;
Contributing authors: nowreen.rafa@gmail.com;
naimur.rahman900@gmail.com; hamimccpc@gmail.com;
islamnazib10@gmail.com; annajiat@gmail.com;
sadiul.arefin.rafi@g.bracu.ac.bd; sania.azhmee.bhuiyan@g.bracu.ac.bd;
†These authors contributed equally to this work.

Abstract

This study dives into the topic of grammatical error correction (GEC), an important area of natural language processing (NLP) that aims to improve textual content by identifying and rectifying grammatical errors. There is a growing demand for automated methods to improve the grammatical quality of text due to the increasing importance of written communication across sectors. Our research introduces novel approaches to improving GEC systems, namely by adapting the gT5 model for use in correcting errors across many languages. Extensive experiments indicate that our method has the potential to significantly improve the quality of written English in a wide variety of settings. The C4_200M GEC dataset, a hand-picked set of 185 million phrase pairings from the clean C4 corpus, is used in the article. Incorporating tagged corruption models, this dataset created by Google researchers provides synthetic training data for GEC that accurately reflects a wide variety of grammatical faults. To demonstrate the potential

of our method in the context of large-scale GEC applications, we conduct analyses on a dataset that has been divided into training, testing, and validation subsets.

Keywords: Fine-Tune, T5 model, Translation, GEC

1 Introduction

When it comes to natural language processing (NLP), grammatical error correction (GEC) is a crucial application with consequences that go well beyond the domain of academia and into everyday technology like word processors, automated essay grading, and language learning applications. Accurate and proper wording cannot be overemphasised, especially in formal writing and other contexts where clarity and accuracy are of the utmost significance. Inaccuracies in syntax, morphology, punctuation, and use can dilute the credibility of a piece of writing and even cause readers to misinterpret its meaning. Therefore, GEC techniques attempt to automatically detect and rectify these problems, enhancing the quality and readability of the text.
Significant progress in machine learning and natural language processing approaches notwithstanding, GEC continues to be difficult. Because of the importance of context and semantics in defining grammaticality, natural languages are notoriously difficult to standardise. Another element of complexity is added by the fact that practical applications typically need fast, low-latency systems that can effectively process massive amounts of text. Modern approaches use a wide variety of methods, such as rule-based systems, statistical models, and, more recently, deep learning strategies. However, there are drawbacks and compromises to every option.
This research paper delves into the domain of GEC with a specific focus on advancing textual error detection and correction. We aim to contribute to this evolving landscape by presenting novel methodologies and techniques for improving GEC systems. By synthesizing ideas from various research avenues, we strive to present a comprehensive framework that addresses the nuances of textual error correction. Our goal is to not only build upon the foundations laid by earlier work but also introduce novel contributions that can further enhance the efficacy of GEC systems. We leveraged the gT5 model [1], a state-of-the-art language model renowned for its text generation prowess, and tailored it to the intricacies of multilingual grammatical error correction. We fine-tuned this already capable model for grammatical error correction and finetuned it on a shortened version of the C4 200M dataset. Our hypothesis was that fine-tuning the model on such an intricate dataset containing multiple nuances of unique error patterns would enable the model to perform even better when it comes it grammatical error correction. As a result, the fine-tuned gT5 model is more sensitive to the subtleties of different languages and it's grammatical errors. It fixes grammar mistakes while taking into account the context and meaning of the text. Through rigorous experimentation and meticulous analysis, we demonstrate the potential of our approach to significantly elevate the quality of written language better than all of the available GEC methods currently prevalent.

2 Literature Review

We studied multiple relevant publications that aligned with our interests for our research. These papers primarily focus on the use of sophisticated neural models, frequently in conjunction with multilingual pre-trained encoders, to capture cross-linguistic patterns and provide reliable corrective performance.

We studied the paper by [1] that introduced an inventive methodology for multilingual grammatical error correction. Their method used a unified neural sequence-to-sequence model with multilingual pre-trained encoders. This model was able to effectively capture shared grammatical structures across languages. Their novel technique eliminated the need for language-specific models and data, streamlining the correction process. Their paper's significance lay in its capacity to address resource limitations associated with individual languages while showcasing competitive performance across diverse languages. By simplifying development and outperforming specific language-specific methods, this paper contributes to the field by offering an efficient and adaptable solution for multilingual grammatical error correction [1].

Additionally, we studied the paper of [2], which used a rule-based approach to generate an error template to match erroneous spans and corrective actions to correct errors. They used a web crawler to extract templates, and the pre-trained language models derived corrective actions. After the extraction and derivation of corrective actions, human evaluation is done to measure their quality. The corrective actions were done by deleting words from left to right, right to left, or randomly selecting a side. The authors of the paper used CTC-2021 datasets, and the P/R/F values demonstrate that the methods used by the authors covered a considerable amount of error.

To overcome the difficulty of teaching models to spot and correct grammatical mistakes, recent developments in Grammatical Error Correction (GEC) have been made. [3] stated that during inference, exposure bias is a problem for conventional models, prompting the frequent use of data augmentation techniques. However, these methods fail to account for the interdependence of different forms of error and lack a sophisticated error repair progression. Type-Driven Multi-Turn Corrections (TMTC) is a ground-breaking method that aims to overcome these constraints. To correct particular sorts of errors, TMTC generates intermediate sentences from training examples. This allows the model to gradually and cohesively learn how to repair errors. The empirical results reveal that TMTC is highly effective, producing state-of-the-art results on English GEC benchmarks [3]. This method not only improves accuracy and memory but also promotes cautious and accurate repair of mistakes. [3] found that its potential to improve GEC models is highlighted by its superior performance in both individual and ensemble models.

Furthermore, we studied the paper by [4], which presented a new method for integrating GEC systems by treating the problem as a simple classification problem. Using only logistic regression, their method achieved significantly better outcomes over state-of-the-art systems on both the CoNLL-2014 and BEA-2019 test sets, with an increase in the F0.5 score of 4.2 and 7.2 points, respectively. What made their approach stand out was that it could be used with any base GEC system. Although Transformer-based architectures were widely used in GEC, there were still notable

differences among the models, especially in terms of problem formulation and pre-training data. This method effectively combined them, capitalizing on the advantages of both. The suggested model, ESC, outperformed previous highly regarded system combination methods without requiring considerable hyper-parameter adjustment in evaluations using CoNLL-2014 and BEA-2019 test data. The research also underscored the potential of integrating syntactic information to enhance GEC systems. Despite the focus on English, the method held promise for other languages given the availability of error-type annotation tools.

The authors of the study, [5], introduced a copy-augmented architecture for the job of Grammatical Error Correction (GEC). This design involved the incorporation of unmodified words from the source phrase into the target sentence. The copy-augmented architecture was pre-trained using a denoising auto-encoder on the unlabeled One Billion Benchmark dataset. A comparison was then conducted between the completely pre-trained model and a partially pre-trained model. Additionally, the authors put forth a more appropriate neural architecture for addressing the grammatical error correction (GEC) problem. This design allowed for the direct inclusion of unmodified words and out-of-vocabulary terms from the source input tokens. The copy-augmented model was initially trained using denoising auto-encoders on a substantial amount of unlabeled data, which effectively addressed the issue of limited labeled training data. Subsequently, the authors assessed the architectural design using the CoNLL-2014 test set, revealing that their methodology surpassed all recently published cutting-edge techniques by a significant margin.

We discovered that these works collectively demonstrated a progression in multilingual grammatical error correction systems, all underpinned by the same objective which was to efficiently capture cross-linguistic patterns.

3 Collected Data

Researchers and practitioners in the field of grammatical error detection (GED) have a substantial problem due to the lack of publicly available datasets. The creation of high-quality datasets is time-consuming and expensive, but it is essential for training and assessing models. This is because it typically requires considerable human annotation by language specialists. The availability of datasets does not guarantee that all grammatical rules and subtleties across languages, genres, or domains will be covered. Effective GED systems require a wide and diversified number of examples to work at their best, and this data deficiency typically limits their development and adoption.

Numerous methods that propose to generate synthetic training data for GEC have been developed as a result of the dearth of adequate training data [6]. Synthetic datasets have become more popular as a solution to this problem. The training data for machine learning models may be improved with the use of synthetic datasets, which are created artificially and can be tailored to contain a broad variety of grammatical mistakes. These datasets help researchers get around problems with sample size and variety, and they also let them account for and manipulate characteristics that may otherwise be difficult to isolate in real-world samples. However, there are still obstacles to overcome when creating synthetic data, such as making fake errors that accurately

reflect those made in the real world. Synthetic datasets, despite these obstacles, are a vital tool for developing the field of grammatical mistake detection, allowing for more robust and generalizable models.

It is well-known that artificial data creation improves the performance of neural GEC systems, but current approaches frequently lack diversity or are too basic to create the extensive variety of grammatical mistakes committed by human writers. The C4 200M dataset was selected because it included error type identifiers from automatic annotation technologies like ERRANT to direct synthetic data production.Based on the massive, well cleaned Common Crawl web crawl corpus (c4). Consequently, we get a brand-new, massive synthetic pre-training data set whose error tag frequency distributions are identical to those of a specified development set. The state-of-the-art has been advanced on the BEA-19 and CoNLL-14 test sets thanks to the use of this synthetic data collection. It was demonstrated that our dataset is superior to high-quality sentence pairings as a means of adapting a GEC system trained on a hybrid of native and non-native English to a test set of native English sentences. We employed a synthetic data set for grammatical error correction that includes several natural-sounding mistakes. This data collection is built on grammatical corruption models, which, when given an error type tag, corrupt an otherwise correct phrase.

To regulate the output of our corruption models and produce more realistic and different grammatical errors, we employ a number of error type tags, such as SPELL (spelling error) and SVA (subject-verb agreement error). By feeding them a clean phrase and an error tag, our tagged corruption models learn to return the corrupted version of the sentence, such as "There were a lot of sheep." becoming "There were a lot of sheeps." Since many error type tags demand more sophisticated rewrites, the tags prevent untagged corruption models from automatically producing oversimplified corruptions. A tidy statement may be transformed into a loud one in many different ways. These synthetic mistakes are often oversimplified when generated by a standard corruption model, but the model may be trained to create error patterns from real-world GEC corpora by using tag information. [7] show that it is beneficial to generate fake data for GEC that encompasses a wide variety of error kinds. Synthetic data can also have its tag distribution adjusted to mirror that of a certain target domain. We employ this distribution matching method to train a GEC system to more effectively fix mistakes made by native speakers. After applying this error distribution to the C4 corpus, the original phrase is used as input and the corrected version is generated as output.

Neural sequence models have a well-known tendency to require large amounts of data, The C4 200M GEC dataset is a collection of 185 million sentence pairs that were derived from the C4 dataset in English. But its primary feature is not its size or diversity, but its cleanliness. The process of cleaning data is a time-consuming one, particularly when one is dealing with the enormity of the web, which contains material that may be redundant, inconsistent, or irrelevant. The "clean" feature of C4 ensures that models have a lower noise-to-signal ratio, which in turn leads to training that is both more effective and meaningful.

4 Data Analysis

The C4 200M dataset was introduced by scholars from Google. So the credibility of the data remains high. Also, the biases in Google-provided datasets are comparatively lower than in other sources. So, the dataset used in this paper is the most suitable for GEC. The dataset was split into three parts: training, testing, and validation. The split was 70%, 20%, and 10%, respectively, for training, testing, and validation. This balanced split of the data ensured proper training and evaluation. The dataset contains 185 million sentence pairs. The sentences are erroneous, and after being passed through the model, the correct grammatical structure of the provided sentences is generated. The sentences are shorter in length, averaging around 9–10 words per sentence, and are written in simple English. As it is the cleaned version of the C4 dataset, the null values are removed from the dataset. Each article in the dataset has two attributes. Input and output. Inputs are the erroneous sentences, and outputs are the corrected sentences with the use of our model. The dataset has been transformed into Parquet format, while older versions of the dataset were available in TSV format. The conversion was prompted by the subpar performance when accessing individual files. I am receptive to receiving requests and comments regarding strategies for effectively managing a large dataset. The dataset is accessible in Parquet format and divided into 10 files, with each file containing around 18 million samples. Each sample consists of a pair comprising an erroneous sentence and its corresponding repaired version. The dataset size was around 16 GB. Processing and training models on large datasets are computationally costly. In this regard, the C4 200M GEC dataset has approximately 185 million sentence records. As a consequence of this, we decided to go with a condensed version of the dataset. This condensed version is known as c4_200m_gec_train100k_test25k, which comprises a total of 125k data, with 100k serving as training data, and 25k serving as test data.

Sample Dataset: {
Input: “I teach first-year accounting and Taxation courses.” Output: “I teach a mixture of first-year Accounting and Taxation courses.” }

5 Methodology

BERT was our first choice to implement out GEC model. But due to the nature of its pre-training, BERT poses special difficulties when applied to grammatical error correction (GEC). In most cases, BERT is taught using a massive database of clean, error-free text. As a result, its structure and weights are calibrated to recognise and produce natural-sounding speech. Since grammatically wrong statements weren’t included in BERT’s training data, the model’s performance may suffer when exposed to them. Keep in mind that while BERT excels in identifying the interconnectedness of words in a phrase, it has no innate grasp of the idea of “grammatical error.” Therefore, it is possible that the best results for GEC will not be achieved by merely fine-tuning BERT on a dataset of grammatically wrong phrases. Creating a custom training regime that includes both correct and erroneous examples is one possible remedy, but without further architectural alterations, a model like BERT may still fail to capture the

intricacies of GEC.So, we decided that BERT may not be as useful for GEC without extensive modifications and looked for other models.

A natural way to think about grammatical error correction (GEC) in the field of natural language processing (NLP) is as a sequence-to-sequence (Seq2Seq) problem. In this scenario, the input sequence is a grammatically incorrect sentence, and the goal is to produce an output sequence that is the revised version of the input sentence. This method treats GEC as a subset of machine translation, where the "source language" is the collection of incorrect phrases and the "target language" is the collection of right ones. Encoder-decoder model architectures are commonly used to tackle the sequence-to-sequence challenge. By reducing the input sentence to a latent "context vector," the encoder is able to capture the sentence's most salient characteristics and interdependencies. Using this context vector as input, the decoder reconstructs the modified phrase token by token. The Transformer model is a well-liked architecture for such jobs because of its impressive performance in many natural language processing tasks. In contrast to more common machine translation jobs, however, GEC as a Seq2Seq issue presents its own set of unique obstacles. Changing verb tenses or altering word order are two examples of the kinds of nuanced changes that call for a sophisticated grasp of syntax and semantics. While making its repairs, the model must also be cognizant of the need to maintain the original sense of the text. Furthermore, in GEC the "source" and "target" languages are essentially the same language but in different levels of grammaticality, whereas in traditional machine translation the source and target languages are separate. Because of this similarity, it might be difficult to teach a model to differentiate between genuine mistakes and variations in style. So, while Seq2Seq models do provide a strong framework for GEC, there are some other factors to think about and maybe architectural tweaks to make to ensure the best performance possible given the specifics of the task.

Given that T5 (Text-to-Text Transfer Transformer) uses an encoder-decoder architecture and the C4 200M dataset is so large, fine-tuning a T5 model on this dataset for grammatical error correction (GEC) seemed like a viable option. However, the model's subpar performance in comparison to specialised GEC models is shown by an F0.5 score of less than 50. The T5 model may not have been sensitive enough to the particular sorts of mistakes that are important to GEC, even after being fine-tuned, and this may explain why it performed so poorly. It's also important to remember that GEC necessitates a finer-grained grasp of syntax and semantics than was initially anticipated for T5. Even while T5 has shown potential in a number of NLP tasks, the raw model was not suitable for a GEC model.

[1] introduced a new model gT5 that produced the best results on the GEC tasks. The gT5 model was built upon the mT5 [8], which has already been pre-trained on a corpus covering 101 languages, is the foundation model the authors used. They used the CLANG-8 dataset. CLANG-8 is the cleaned version of the widely used LANG-8 dataset. The authors of the paper found remarkable results with the CLANG-8 dataset on their gT5 model. Now, we propose to improve the results obtained by [1]. To achieve this, we fine-tuned the C4 200M dataset on their gT5 model. Our model is built on top of gT5 [1], a multilingual version of T5 [9], a Transformer encoder-decoder model that has been demonstrated to produce cutting-edge outcomes on a variety of NLP

tasks. As C4 200M consists of about 185 million sentence pairs, it consists significant amount of sentences for the model to train. But fine-tuning 185 million sentences is computationally very difficult. We also didn't have the resources to complete that in the limited time. That's why we used 125k sentence pairings for our project. Among them, 100k are for the training set and 25k are for the testing set. The fact that gT5 was trained on paragraphs rather than individual sentences is another limiting factor. As we are using sentences to correct errors, our hypothesis is that it will improve the performance by a great extent. We followed the same processes and steps to fine-tune the gT5 model that is followed for fine-tuning the sequence-to-sequence translation models. The input is prefixed with a prompt to indicate to T5 that the task at hand is a translation task. Certain models that possess the ability to do various natural language processing (NLP) tasks necessitate the use of prompts tailored to the unique tasks at hand. The input (incorrect sentences) and target (correct sentences) should be tokenized using a tokenizer that has been pre-trained on an English vocabulary. To ensure that sequences do not exceed a certain maximum length, it is necessary to truncate them accordingly. We set the learning rate to be 2^-5. We set the batch size for training to be 16 per device and 16 per device for evaluation. The weight decay was set to 0.01. The total save limit was set to 3. We ran our code for two epochs due to our shortage of high-end devices. Finally, we pushed our code into huggingface in the name "gsg-T5model". Although it has been demonstrated that using synthetic data as the first step in fine-tuning increases model accuracy, using synthetic data also introduces practical issues that make the development and fair comparison of GEC models difficult.

6 Result Analysis

Through analysis we found out that our model was capable of fixing grammatical error in most cases without needing to eliminate or add new words. For example, for an input "I like to swimming" the gT5 model outputs "I like swimming" but our gsg-T5 model outputs "I like to swim". This is an already improvement over the gT5 model. Then we compared the F0.5 score of the two models.

The F0.5 score is a subset of the F1 score, a popular statistic for gauging a classification model's efficacy. The F0.5 score lays more emphasis on accuracy than the F1 score does, which weighs both precision and recall equally. It excels in situations where the cost of a false positive is higher than that of a false negative, or if it is more important to accurately identify occurrences than to do so completely. Tasks like grammatical error correction benefit from this measure since inaccurate repairs might lead to new errors or change the original meaning, respectively.

At first, we performed F0.5 score on C4 200M dataset. We picked the next 20,000 sentences not included in the test or train set. Then passed the input (incorrect grammatical sentence) into the gT5 model and the gsg-T5 model. We compared the results with the output column of the C4 dataset. Then we calculated the F0.5 score, gT5 got an F0.5 score of 44.10 and out model got 47.10. This is a remarkable improvement considering we only used 100K test sentences to fine-tune the model. Then we evaluate on standard benchmarks from CoNLL-14 (noalt)5 and the BEA test [10], F0.5

Scores. We used the M2 scorer for CoNLL-14, Russian, Czech and German, and the ERRANT scorer [10] for BEA test. The CoNLL-14 Shared Task is used to measure how well GEC programmes work. It gives a standardised set of text with spelling mistakes that is often used to test how well different error correction models work. In the same way, the BEA (Building Educational Applications) Test is another well-known way to evaluate GEC. The BEA test sets tend to cover a wide range of types of mistakes, such as lexical, grammatical, and style errors, but not just those. Because of this, the BEA test is a complete way to evaluate GEC models in all areas of language repair. The results of the test is

Table 1 Performance Comparison on BEA Test and CoNLL-14

Model	BEA Test (F0.5 Score)	CoNLL-14 (F0.5 Score)
gT5	60.2	54.10
gsg-T5	63.1	56.01

References

[1] Rothe, S., Mallinson, J., Malmi, E., Krause, S., Severyn, A.: A Simple Recipe for Multilingual Grammatical Error Correction (2022)

[2] Zhang, Y., Jiang, H., Bao, Z., Zhang, B., Li, C., Li, Z.: Mining Error Templates for Grammatical Error Correction (2022)

[3] Lai, S., Zhou, Q., Zeng, J., Li, Z., Li, C., Cao, Y., Su, J.: Type-Driven Multi-Turn Corrections for Grammatical Error Correction (2022)

[4] Qorib, M.R., Na, S.-H., Ng, H.T.: Frustratingly easy system combination for grammatical error correction. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1964–1974. Association for Computational Linguistics, Seattle, United States (2022). https://doi.org/10.18653/v1/2022.naacl-main.143 . https://aclanthology.org/2022.naacl-main.143

[5] Zhao, W., Wang, L., Shen, K., Jia, R., Liu, J.: 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), pp. 156–165. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1014 . https://aclanthology.org/N19-1014

[6] Lichtarge, J., Alberti, C., Kumar, S., Shazeer, N., Parmar, N., Tong, S.: 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), pp. 3291–3301. Association for Computational Linguistics, Minneapolis, Minnesota (2019). https://doi.org/10.18653/v1/N19-1333 . https://aclanthology.org/N19-1333

[7] Wan, Z., Wan, X., Wang, W.: Improving grammatical error correction with data augmentation by editing latent representation. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 2202–2212. International Committee on Computational Linguistics, Barcelona, Spain (Online) (2020). https://doi.org/10.18653/v1/2020.coling-main.200 . https://aclanthology.org/2020.coling-main.200

[8] Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., Raffel, C.: mT5: A massively multilingual pre-trained text-to-text transformer (2021)

[9] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (2020)

[10] Bryant, C., Felice, M., Andersen, Ø.E., Briscoe, T.: The BEA-2019 shared task on grammatical error correction. In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 52–75. Association for Computational Linguistics, Florence, Italy (2019). https://doi.org/10.18653/v1/W19-4406 . https://aclanthology.org/W19-4406