Finetuning LLMs for Grammatical Error Correction in English and Greek Texts

Dionisios Kapelles
University of Patras
Patras, Greece
st1067479@ceid.upatras.gr

Andreas Andriopoulos
University of Patras
Patras, Greece
andriopa@ceid.upatras.gr

Dimitrios Koutsomitropoulos*
University of Patras
Patras, Greece
kotsomit@ceid.upatras.gr

Abstract

Accurate grammatical correction is paramount for effective communication, especially for non-native speakers of a language. This research aims to harness the power of the generative capabilities of LLMs, such as the T5 model and transfer learning to develop an efficient and flexible system for automated grammatical correction in written text. It involves the finetuning of a pre-trained T5 model on a custom dataset containing English sentences with varying degrees of grammatical errors. Results show the effectiveness of the improved T5 model in grammatical correction. We also report results for a low-resourced scenario of Greek texts, using the multilingual model's variant. The model achieves competitive performance on benchmarking metrics, outperforming existing methods in terms of accuracy and contextual understanding. The findings highlight the importance of using pre-trained models and finetuning techniques to develop sophisticated grammar correction systems and writing aids.

Keywords

GEC, pretrained model, T5, JFLEG, fine-tuning

ACM Reference Format:
Dionisios Kapelles, Andreas Andriopoulos, and Dimitrios Koutsomitropoulos. 2025. Finetuning LLMs for Grammatical Error Correction in English and Greek Texts. In *The PErvasive Technologies Related to Assistive Environments (PETRA '25), June 25–27, 2025, Corfu Island, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3733155.3733222

1 Introduction

Understanding the peculiarities of grammar and syntax remains a challenge, both for native and non-native speakers of a language. Grammatical errors in written texts can impede comprehension and reduce the overall quality of content. Recognizing the widespread presence of these errors and their impact on communication, the field of grammar error correction (GEC) has evolved over time. Numerous approaches to grammar correction have been explored ranging from rule-based systems and statistical models to machine translation [14], for example, GEC applications found in word processors, web-based writing services, such as Grammarly, etc.

In this paper we aim to harness the power of the generative capabilities of the T5 (Text-to-Text Transfer Transformer) model [3] to develop an efficient and flexible system for automated grammar correction in written text. The proposed methodology involves finetuning a pre-trained T5 model on a customized dataset containing sentences with varying degrees of grammatical errors. The model employs the sequence-to-sequence function, allowing it to produce corrected sentences for input with grammatical errors. In addition, we investigate its performance on a low-resources scenario involving custom made datasets of Greek texts, thus accounting for the morphological complexities of mid- to low-resourced languages, such as Greek. It is shown that finetuning can perform well even in the absence (or total lack thereof) any specific language pretraining.

We also comparatively evaluate the model against other current LLMs and ensemble approaches and show it achieves competitive performance on evaluation metrics, outperforming existing methods in terms of accuracy and contextual understanding. The present study provides insights into the strengths of the model, its ability to handle diverse error patterns, and areas for further improvement.

To our knowledge, there is no systematic review and evaluation of finetuning language models for the specific task of grammar correction, while experiments have been carried out on a low-resourced Greek dataset using both T5 and its multilingual variant, mT5 [10] and evaluated with the GLEU metric [6]. Our source code is openly available at: https://github.com/DennisKapelles/Grammatical-error-correction-in-JFLEG-dataset-with-T5-text-to-text-transformer .

The rest of this paper is organized as follows: in Section 2 we review relevant literature in the field of grammatical error correction; in Section 3 we present our methodology and approach, by outlining the error correction procedure designed with the T5 model and the adaptation to the Greek language. Section 4 summarizes the evaluation configuration, key metrics and loss function used. Section 5 contains the results of our experiments and their analysis as well as a comparative evaluation with current state-of-the-art. Finally, section 6 outlines our conclusions and future work.

2 Related Work

Several studies have shown that GEC can be approximated with machine translation using a Seq2Seq model [16] along with more recent Transformer architectures [18]. Previous work has utilized various advanced models. Transformer-based models [11], such as ours, leverage the classical Transformer architecture [1] which excels at sequence-based tasks due to its self-attention mechanisms that capture long-term dependencies within sentences. Another approach combines Statistical Machine Translation (SMT) with a BiGRU network [13], enabling the model to understand context

[*] Corresponding author

PETRA '25, Corfu Island, Greece
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1402-3/2025/06
https://doi.org/10.1145/3733155.3733222

Table 1: A sample of the JFLEG dataset with data.

sentence	Corrections
I want to talk about nocive or bad products like alcohol , hair spray and cigarrets .	[”I want to talk about nocive or bad products like alcohol , hair spray and cigarettes . ”, ”I want to talk about harmful or bad products like alcohol , hair spray and cigarettes . ”, ”I want to talk about harmful or bad products like alcohol , hair spray and cigarettes . ”, ”I want to talk about harmful or bad products like alcohol , hair spray and cigarettes . ”]
For not use car .	[”Not for use with a car . ”, ”Do not use in the car . ”, ”Car not for use . ”, ”Can not use the car . ”]
There are several reason .	[”There are several reasons . ”, ”There are several reasons . ”, ”There are several reasons . ”, ”There are several reasons . ”]
Thus even today sex is considered as the least important topic in many parts of India .	[”Thus , even today , sex is considered as the least important topic in may parts of India . ”, ”Thus , even today , sex is considered the least important topic in many parts of India . ”, ”Thus , even today , sex is considered the least important topic in many parts of India . ”, ”Thus , even today sex is considered as the least important topic in many parts of India . ”]
For example they can play football whenever they want but the olders can not .	[”For example , they can play football whenever they want , but the elders cannot . ”, ”For example , they can play football whenever they want but the others can not . ”, ”For example , they can play football whenever they want but the seniors ca n't . ”, ”For example they can play football whenever they want but the older ones cannot . ”]

from both directions of a sequence, enhancing its corrective capabilities. The copy-augmented model [17] introduces a ”copy mechanism” that allows for directly copying segments from the input to the output, which is particularly useful in GEC since many words and phrases remain unchanged in corrections. Additionally, Transformer models pre-trained with pseudo data and enhanced with BERT-based model [12] improve language understanding and the quality of generated corrections. The tagged corruptions model [7] introduces intentional ”errors” in the training data, helping the model to recognize and correct standardized mistakes within sentences. VERNet [19] is a specialized model for GEC that uses tailored approaches to handle complex language structures and grammatical errors more effectively. Finally, CNN Seq2Seq model [2] leverage Convolutional Neural Networks for sequence-to-sequence transformations, with Seq2Seq architecture used to learn the relationships between original and corrected sentences.

The T5 model has been effectively applied in several grammatical error correction tasks and related language processing projects. For example, in GECToR [9], T5 was used as a reference model to compare the effectiveness of tagging versus text-to-text correction for error correction. T5 has also been applied for correcting mistakes made by second-language learners, demonstrating T5's adaptability to varied language structures. In the BEA 2019 Shared Task [3], T5 was utilized by various research teams for GEC, leveraging its text-to-text structure for accurate error correction. Finally, in [15] authors combine T5 with data augmentation techniques to enhance GEC performance, particularly for low-resource datasets. These projects showcase T5's flexibility and efficiency in addressing grammatical correction and other complex linguistic tasks.

To account for the morphological complexities of the Greek language, a related effort is presented in [8], which introduced methods to enhance GEC resources specifically for Greek. This study used datasets built from Greek learner corpora and evaluated the models using metrics such as Precision, Recall, and F0.5. The best-performing model, a transformer-based architecture, achieved an F0.5 score of approximately 60%, indicating strong performance in prioritizing precision over recall in GEC tasks for Greek.

3 Methodology

3.1 Dataset

A JFLEG dataset is used to train the model. JFLEG (JHU Fluency-Extended GUG) is a dataset for grammatical error correction in English [5]. It is a gold reference standard for developing and evaluating GEC systems in terms of fluency (whether a text resembles its native language) as well as grammar. For each original text (sentence), there are four corrections written by observers (corrections) following specific guidelines (Table 1).

Each instance contains a source sentence and four corrections. Sentence field contains the original sentence written by an English learner and corrections field contains corrected versions by human annotators. The order of the annotations is consistent (eg first sentence will always be written by annotator ”ref0”). The dataset contains a total of 1,503 records and is divided into two roughly equal parts: validation (755 rows) and test (748 rows). The dataset is available on the Hugging Face dataset provider network and can be accessed using the Datasets library.

3.2 Models Used

The T5 model, developed by Google, is a transformer-based language model designed to handle a wide range of natural language processing (NLP) tasks in a unified ”text-to-text” format. This means that it takes both input and output as text strings, making it adaptable to tasks such as translation, summarization, question answering, and grammatical error correction. T5 was trained on the large-scale C4 dataset and is notable for its flexibility, enabling fine-tuning for diverse NLP applications while maintaining strong performance across tasks. In this task, we use the pre-trained transformer model t5-base, the variant of the T5 model, which is considered the basic model for such treatments and has about 220 million parameters.

A multilingual variant of T5, mT5 (multilingual T5) [10], can be used for Greek datasets, as it has been pre-trained on data from many languages, including Greek. mT5 was pre-trained on a dataset covering 101 languages, following the paradigm of the more general solution of pre-training on multiple languages. Greek ranks 20th in data representation, with 43 billion tokens, making mT5 well-suited for Greek NLP tasks. The least common languages were Sotho and Yoruba, and other indigenous languages that have a much smaller amount of data available. In our task, we use one of the variants of the mT5 model, mT5-base, which contains 580 million parameters.

Both T5 and mT5 use an Encoder-Decoder transformer which is pre-trained with masked language modeling, covering successive intervals of input tokens and then attempting to reconstruct them. T5 was pre-trained on 750 GB of English-language text derived from the public web Common Crawl. mT5 was pre-trained on data from all 71 monthly web data published by Common Crawl so far, which is more than the source data used by T5.

3.3 Implementation

We use a Python package, Happy Transformer, to implement our work. Happy Transformer, built on top of Hugging Face's Transformers library, facilitates fine-tuning and inference with NLP Transformer models, as well as implementing and training transformer models.

The process begins with loading the pre-trained model t5-base. Before we proceed with training the model, we need to preprocess our dataset, to correct any errors it contains and improve the performance of the grammar correction. Some of the instances within the dataset contain too many blanks and if not corrected, the model will produce blanks when not required.

The next part of the preprocessing is to bring the dataset into the appropriate format for Happy Transformer. We need to structure both the training data and the evaluation data in the same format, which is a CSV file with two columns: input and target. The input column contains grammatically incorrect text (source) and the target column contains text corresponding to a single corrected sentence for each one of the four in JFLEG. In total, we create 3,016 training examples and 2,988 evaluation examples. After having finished all the preprocessing, we are ready to fine-tune our pre-trained transformer model. A summary of the implementation workflow is presented in Fig. 1.

For our task we need to specify which procedure we want to perform by adding the same prefix to each input. In this case, we will use the ”grammar:” prefix. This is because T5 models are able to perform multiple tasks, such as translation and summary, with a single model and a unique prefix used for each task, so that the model knows which task to perform.

3.4 Adaptation to Greek

As with the T5 model used for the English texts, we follow the same procedure and technique for the mT5 model for the Greek texts. At the same time, we perform experiments with Greek texts against the original T5 (t5-base) to showcase ablation. To account for the low-resourced Greek scenario, we use two manually built datasets, one for fine-tuning the model and one for evaluating it, with 50 incorrect sentences and their corresponding corrections each. These datasets follow the patterns that the mT5 model accepts as input and are sufficient for its correct operation. The dataset structure is similar to that of JFLEG: it consists of two columns containing incorrect sentences with their corresponding corrections.

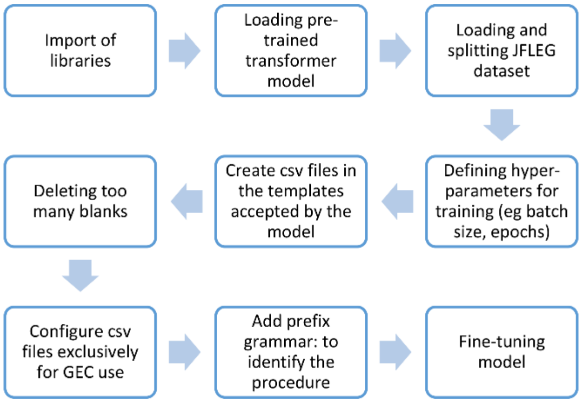
Import of libraries
Loading pre-trained transformer model
Loading and splitting JFLEG dataset
Defining hyper-parameters for training (eg batch size, epochs)
Create csv files in the templates accepted by the model
Deleting too many blanks
Configure csv files exclusively for GEC use
Add prefix grammar: to identify the procedure
Fine-tuning model

Figure 1: Implementation workflow.

We have introduced specific types of errors to tailor our dataset to the specific morphology of the Greek language. For example, we have added errors of accent, as accent in Greek is indicated by a punctuation mark (') and not just by the emphasis of the word during speech. In addition, errors have been added concerning the final n. The retention or omission of the final n in certain Greek words (articles, pronouns or particles), because of its frequency as an error even by native speakers, is considered a different type of error and not simply a spelling error. Sample contents of the two datasets are presented in Table 2.

4 Evaluation

4.1 Configuration

Our work is carried out in the Google Colab environment. This environment comes with 16GB memory and uses GPU (T4 GPU) for hardware acceleration, which helps due to the large amount of data that needs to be generated for the task.

The hyper-parameters defined for fine-tuning T5 are: the batch size equal to 8 and the epochs equal to 10. In addition, we use a *default learning rate* equal to 5e-5, *max_input_length* and *max_output_length* equal to None and *fp16* equal to False. *Max_input_length* and *max_output_length* reflect the maximum number of tokens for input and output. The rest are truncated. By default, the maximum number of tokens the model can handle is used. If *fp16* is true, it enables half-precision training, which saves space by using 16 bits instead of 32 to store the model weights. The loss function used is Cross-Entropy Loss. Token-level losses are summed or averaged over the entire output sequence to obtain the total loss for that sequence.

For finetuning mT5, we use the mt5-base model. We use batch size and epochs as previously and the default learning rate of 3e-5, *min_source_length* equal to 256 as the *max_target_length*, which are adequate for sentence size.

Table 2: Sample contents of the two Greek datasets (training and evaluation)

Sentence	Corrections
Εχωέναβιβλίο .	Έχωέναβιβλίο.
Αυτηείναιτισπιτιμου .	Αυτήείναιτοσπίτιμου.
Εμειςπαμαιστοπαρκοκαθεμερα .	Εμείςπάμεστοπάρκοκάθεμέρα.
Αυτοιειναιοικαλυτερηφιλημου .	Αυτοίείναιοικαλύτεροιφίλοιμου.
Το σκυλοςειναιμεγαλοκαιγρηγορο.	Το σκυλίείναιμεγάλοκαιγρήγορο.

Table 3: Evaluation results on JFLEG.

Stage of implementation	GLEU score	Loss score
t5-base before fine-tuning	0.0945	1.28039
t5-base after fine-tuning	**0.7764**	**0.47939**

4.2 Metrics

GLEU [6] is a variant of the BLEU metric, adapted for evaluating GEC systems. It measures the n-gram overlap between corrected proposals and reference proposals, taking into account both the accuracy and recall of the n-grams. Accuracy measures the percentage of corrections made by the GEC system that are actually correct. High accuracy indicates that the model makes few incorrect corrections. Recall measures the percentage of actual errors in the text that the GEC system detects and corrects correctly. High recall indicates that the model detects most errors. GLEU provides a balanced view of the performance of the GEC system, taking into account both the fluency and adequacy of the corrections. It is particularly useful for capturing subtle grammatical refinements that may be missed by other metrics.

GLEU usually ranges between 0.0 and 1.0. If the two sentences are perfectly matched, then GLEU = 1.0. Conversely, if the two sentences do not match at any point, then GLEU = 0.0. The core of the GLEU evaluation metric is the detection of the number of words of common occurrence between the hypothetical sentences and the reference sentences.

5 Results and Discussion

5.1 T5 fine-tuning results on the English dataset (JFLEG)

In Table 3, we report GLEU and loss before and after the fine-tuning of our pre-trained model. We observe that the model loss is significantly high before training. This means that the model is not yet able to correctly predict the expected outcome, i.e. the correct sentences. The loss after training is significantly reduced. This means that the model is trained correctly, and we have achieved the optimization we are looking for.

We use a list of sentences and the T5 model to generate corrected sentences from the uncorrected ones. Now that we have the candidate sentences (predictions) available, we can calculate the GLEU score before and after the fine-tuning. As we can see, the value of the metric is quite higher than the one before fine-tuning, which indicates that our model mostly produces correct results. Approximately 77 out of 100 suggestions produced by our model agree with the target suggestions of the evaluation dataset.

5.2 Comparative Evaluation

Table 4 shows the GLEU scores of the various models used for GEC and reviewed in related work (Section II), on JFLEG-type datasets as reported in the literature. We observe that using a transformer-based model such as ours, we achieve a performance of about 60%. Our model shows higher performance than other models of the same type and, in fact, outperforms other approaches, possibly due to the increased model size as well as the specific finetuning on JFLEG.

Moreover, we see that using a combination of models SMT + BiGRU increases the GLEU score by about 2%. Model combinations that contain their own unique features are likely to cover or even correct imperfections that the models may exhibit individually. Grammatical error correction using a Transformer-based model ranks 7th in the GLEU score ranking table, while grammatical error correction using a Transformer + Pre-train with Pseudo Data + BERT-based model ranks 4th. In conclusion, the tagged corruptions model and the VERNet model achieve the highest performances compared to the previous works, but their GLEU score values are still lower than the one presented here, while the CNN Seq2Seq model achieves the lowest performance among all the models.

5.3 mT5 and T5 finetuning results on the Greek dataset

Following the same steps, we calculate the loss and the GLEU score of the mT5 model before fine-tuning, fine-tune the mT5 model on the Greek training dataset and then calculate the new loss and evaluate it using the GLEU metric on the Greek evaluation dataset. As a means of ablation, we repeat the same experiment with the original T5, which has not been pretrained with Greek texts. The results obtained are shown in Table 5.

Fig. 2 and 3 show the loss curves of the mT5 and T5 models during their fine-tuning on the Greek dataset. By observing the loss before and after fine-tuning, we understand that although the loss of the model is reduced, it remains at very high levels. This is due to the very small data sample we provide to the model, as on a much larger dataset it would show better results. We observe the same in the GLEU score for our model. Similarly to before, the GLEU score increases after fine-tuning but remains at very low levels, around 42%.

In Table 5 we observe that the mT5 model performs better compared to the T5 model, which indicates that mT5 is more suitable

Table 4: GLEU score of our task compared to scores of models applied to JFLEG dataset

Rank	Model	GLEU score
1	Our task of GEC with T5	**77.6**
2	Tagged corruptions [7]	64.7
3	VERNet [19]	62.1
4	Transformer (EncDec architecture) + Pre-train with Pseudo Data + BERT [12]	62.0
5	SMT + BiGRU [13]	61.5
6	Copy-augmented Model (4 Ensemble + Denoising Autoencoder) [17]	61.0
7	Transformer (self-attention-based model) [11]	59.9
8	CNN Seq2Seq [2]	57.47

Table 5: Evaluation results of mT5 and T5 models on the Greek dataset

Model	GLEU score before fine-tuning	GLEU score after fine-tuning	Loss score before fine-tuning	Loss score after fine-tuning
mT5	0.0715	**0.4226**	1.41930	**1.09475**
T5	0.0378	0.3106	1.73729	1.38173

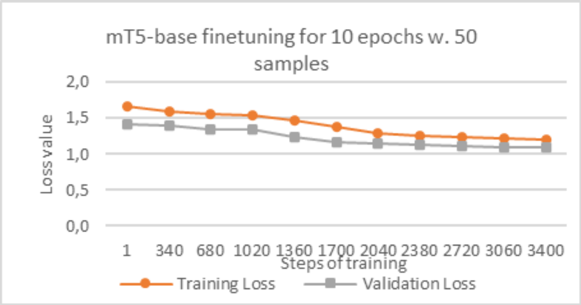

Figure 2: Curves of training and validation losses for the mT5 model

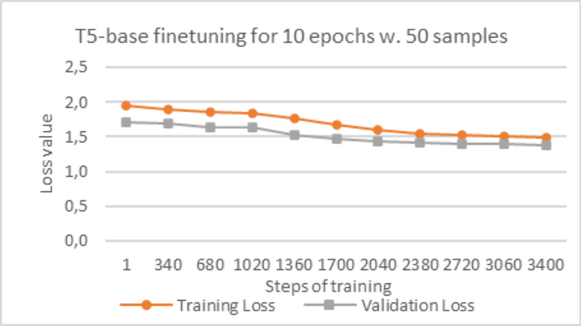

Figure 3: Curves of training and validation losses for the T5 model

for Greek language tasks, due to the multilingual training it has received. However, when low resources are the case, both in terms of available data as well as computational power, finetuning seems to pay off (Table 5): even with only 10 epochs and 50 samples, the original T5 of 220M outperforms the much more expensive 580M mT5 although pretrained on 43B Greek tokens.

6 Conclusions and future work

Our work shows satisfactory results in correcting grammatical errors and exceeds the score of many works in previous years, while approaching the levels presented by other models (BERT, GPT, hybrid approaches). This is indicative of the capability of fine-tuning to adapt the model and means that our model is trained correctly and produces correct results.

Dataset size and quality in a low-resources scenario are also shown to be important. The volume of parallel data in GEC is not comparable to even the largest Lang-8 dataset (1,147,451 sentences). The size of the data corpus in languages other than English is significantly smaller. As indicated by this work, a feasible solution may come from carefully fine-tuning existing language models; whenever possible, pre-training and data augmentation strategies can help to incorporate large amounts of error-free text for low-resource languages.

Although adequate for finetuning, our work uses a Greek dataset that is relatively and purposedly small. A larger dataset might be useful, especially when representing greater GEC domains (native speakers, foreign learners etc.). Especially for L1 (the author's first language), treating as equivalent texts written by authors with different first languages causes small returns on evaluation metrics (F0.5) approaching 50%.

Comparison and relation to even larger and generative LLMs such as GPT, Llama, Gemini and Claude could be investigated. Few-to zero-shot learning can be leveraged to analyze and compare their performance on GEC, with a particular focus on languages they are not pre-trained with.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. 2017. Attention Is All You Need. Advances in Neural Information Processing Systems 30.

[2] Chollampatt, Shamil & Ng, Hwee. 2018. A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction. Proceedings of the 32nd AAAI Conference on Artificial Intelligence.

[3] 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, pp. 52-75.

[4] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. J. Mach. Learn. Res., 21, 140:1-140:67.

[5] Courtney Napoles, Keisuke Sakaguchi, Joel Tetreault. 2017. JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 229–234.

[6] 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.

[7] 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.

[8] Katerina Korre, John Pavlopoulos. 2022 Enriching Grammatical Error Correction Resources for Modern Greek. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 4984–4991.

[9] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, Oleksandr Skurzhanskyi. 2020 GECToR – Grammatical Error Correction: Tag, Not Rewrite. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 163–170.

[10] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 483-498.

[11] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task. 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

[12] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, 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, pp. 4248–4254

[13] Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 284–290, New Orleans, Louisiana. Association for Computational Linguistics.

[14] Sagar Ailani, Ashwini Dalvi, Irfan Siddavatam. 2019. Grammatical Error Correction (GEC): Research Approaches till now. International Journal of Computer Application, 178(40), 1-3.

[15] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, Kentaro Inui. 2019. An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction. In 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, pp. 1236-1242.

[16] Thang Luong, Hieu Pham, Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421

[17] 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.

[18] Ying Zhang, Hidetaka Kamigaito, Manabu Okumura. 2023. Bidirectional Transformer Reranker for Grammatical Error Correction. Journal of Natural Language Processing, 31(1), 3-46.

[19] Zhenghao Liu, Xiaoyuan Yi, Maosong Sun, Liner Yang, and Tat-Seng Chua. 2021. Neural Quality Estimation with Multiple Hypotheses for Grammatical Error Correction. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5441–5452, Online. Association for Computational Linguistics.