

A Comparative Review of Grammatical Error Correction Techniques: Statistical, Neural, and Transformer-Based Approaches

Anwar Aldahan, Ohoud Alqria, Fatimah Alwarsh, Shahad Alshehab

Abstract—Grammatical Error Correction (GEC) is a fundamental activity of Natural Language Processing (NLP), serving in the tools of writing assistant, education, and language assistance of non-native speakers. GEC studies have progressed tremendously since the initial statistical and rule-based systems to neural networks and, most recently, transformer-based models that provide high-fluency context-specific corrections. The given paper includes a comparative analysis of four successful GEC strategies, namely GECToR, T5-based multilingual GEC, Copy-Augmented Pre-Training, and Multi-Task Optimized Transformer Models. We contrast the performance of models in terms of precision, fluency and F0.5 results using benchmark datasets; on FCE, CoNLL-2014, BEA-2019 and JFLEG. The results indicate that higher generating models are always better than previous techniques; GECToR is much faster when using tags to correct pictures and T5 gives greater generative power. Copy-augmented and multi-task models are effective in covering errors, although there is still an issue of over-correction and weakly-resource behaviour. This paper draws conclusions on existing shortcomings and future research avenues to the development of strong, scalable and multilingual GEC systems.

Index Terms—Grammatical Error Correction (GEC), Natural Language Processing (NLP).

I. INTRODUCTION

RECENTLY, Natural Language Processing (NLP) has been a key focus in the current artificial intelligence, which allows machines to learn, comprehend and synthesize human speech. Its applications include machine translation, text discerning, conversationalist aids, plagiarism trackers, and machine aiding writing assistants. The Grammatical Error Correction (GEC) is one of the NLP tasks that are most significant and that concentrates on the automatic detection and correction of grammatical, spelling and fluency errors in a text. GEC is specifically useful in language learners, scholarly writing, content generation platforms, and assistive technologies in writing where a generation of texts in accurate and fluent text is needed. GEC has experienced changes over time; initially it relied on statistical and rule-based systems, which were highly reliant on a set of pre-defined linguistic regularities, then neural compute systems where sequence-to-sequence learning can happen, and most recently transformer-based systems and pretrained language models, including BERT, T5, and GECToR. Every successive generation of techniques has

Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia.

Emails: {2220004414, 2220005404, 2220003876, 2220003900}@iau.edu.sa

brought improvements in accuracy of correction, ability to generalize as well as scalability, however, there are still issues with rare types of mistakes, fluent corrections without over-editing and high results with low-resource languages. There are a variety of datasets and benchmarks corpora that can be used in GEC research, including FCE, JFLEG, CoNLL-2014, and BEA-2019, which can be freely accessed on platforms like HuggingFace and Kaggle as well as open research repositories. These tools have facilitated large-scale analyses based on measures such as M2 score, ERRANT, Precision / Recall/F-score and comparison of models among models are possible and relevant. This work gives a comparative analysis between key GEC methods, including innovations in statistical and neural models, transformer-based and their advantages, disadvantages, and developmental trend. The focus of the aim is to offer a systematic explanation of how GEC techniques have evolved in the course of time, what are the existing restrictions and where research is moving to in future.

II. BACKGROUND

GEC is one of the most significant Natural Language Processing (NLP) topics that has been evolving over the decades. This section provides the linguistic and conceptual background to the methodologies which are to be examined in the subsequent sections of the paper such as the classification of the errors, conceptual task formulation and GEC development history.

A. Types of Grammatical Errors

The term grammatical is usually employed in loose meaning but the GEC task is multifaceted because the errors that can be corrected are very many and provide various linguistic errors such as syntax, morphology, semantics and even fluency. As stated in the recent literature [1], [2], the errors that GEC systems are used to address usually fall into the following general categories:

- Syntactic Errors: Misorder of words, structure or alignment of sentence construction (e.g. omitted subject/verb). Indicatively, certain construction patterns can be used to implement corrections on complex syntactic structures [3].
- Morphological Errors: These are errors in words, tenses, and agreement (e.g., subject-verb agreement, noun plurals, verb form, etc.).

- Lexical Errors: Applying words wrongly (e.g. homophones) or using common expressions improperly.
- Orthography (Spelling): Simple spelling mistakes, typing mistakes and error in punctuation.
- Fluency/Style: Edits that improve naturalness, coherence or style of the text, and do not typically abide by the rules of correct grammar [2].

The main problem with this area is that GEC systems have to contend with all these forms of error concurrently. Recent studies have noted the advantage of using the linguistic patterns, including Construction Grammar (CxG), to capture the underlying language use and provide remedies to the more complex syntactic errors such as missing or redundant words [2].

B. GEC as a Sequence-to-Sequence Task

In its modern application, GEC is a sequence-to-sequence (Seq2Seq) or a text-to-text task. This framing transformed the field by making mistake finding (a classification problem) and mistake fixing (a generation problem) the new focus, with heavy reliance on new inventions in Neural Machine Translation (NMT) [2].

A sequence-to-sequence model considers the input sentence (Source Sequence, S) to be ungrammatical, and the goal of the model is to learn the function of mapping the ungrammatical input sentence (Source Sequence, S) to the grammatical output sentence (Target Sequence, T) [4]. Although the pure Seq2Seq paradigm (i.e. T5) can directly produce the fixed sentence, numerous state-of-the-art GEC models can use the Seq2Edit paradigm [5]. The correction problem is transformed into a series of particular edit operations (INSERT, DELETE, REPLACE) on the source sentence by Seq2Edit, which results in greater inference efficiency [3].

C. Key NLP Concepts

The contemporary GEC is built on a series of ideas, invented during the age of the NLP and deep learning:

- Language Models (LMs): This is a statistical or neural model giving a probability distribution to a sequence of words. Modern GEC uses Large Language Models (LLMs) such as T5 and GPT, which are trained on large scale datasets to find grammatical and semantic coherence [6], [5].
- Embeddings: Embeddings are dense vectors of words, subwords or characters, which are low-dimensional. Embeddings describe semantic and syntactic connections, and words that are similar in senses (or roles) will be given similar vectors [7].
- Transformer architecture: It was an alternative architecture to recurrent models (RNNs/LSTMs), introduced in 2017. Its primary component is the Self-Attention mechanism that allows the model to score the importance of all the words of the input, and that of the one one is working with, effectively accessing long-range dependencies [6].

D. Historical Perspective of GEC

The history of GEC systems development also reflects the overall history of the NLP in that it evolved through three principal paradigms [8]:

- 1) Rule-Based Era (Pre-2000s): Rules The initial GEC systems were based on hand coded rules and special lexicons. Rule-based systems are still used even more recently and are still applicable to high-precision tasks in specialized fields. As an illustration, Gunter et al. [8] confirmed a rule-based NLP algorithm to extract stroke data with high specificity, and Qian et al. [9] showed in 2025 that, due to its interpretability, a deterministic rule-based algorithm can outperform general models to recognize clinical events, such as falls.
- 2) Statistical Era (2000s-2010s): This is the period when statistical models, such as Naive Bayes and Statistical Machine Translation (SMT), were implemented because of the emergence of machine learning. These methods acquired error information directly on the basis of parallel corpora, which is more robust than its predecessors based on rules [2].
- 3) Neural Era (2014-Present): Techniques of deep learning allowed significantly enhancing the quality of GEC. Transformers and Pre-trained Language Models (PLMs) are the current state-of-the-art [6]. Nonetheless, there are issues facing the low-resource languages, where recent studies consider implementing those models using languages which have sparse training resources, including Zarma [10].

E. Project Goals and Objectives

The main objective of the research is to perform systematic and comparative analysis of evolutionary path of Grammatical Error Correction (GEC) systems. This research is aimed at synthesizing the findings of the last decade, unlike empirical studies, which are concerned with the creation of one architecture. In particular, this project has fourfold objectives:

- 1) Taxonomic Classification: To systematically group GEC methodologies into three different eras namely: Statistical/Rule-Based, Early Neural (RNN/CNN), and Transformer-Based), and emphasizes the technological factors underlying each change.
- 2) Performance Benchmarking: To strictly compare the performance of the most state-of-the-art models, in particular, whether to compare Sequence-to-Edit models (e.g., GECToR) and Sequence-to-Sequence generative models (e.g., T5). It includes the examination of performance in terms of standard measures (Precision, Recall, F0.5) on well-known data sets including CoNLL-2014, BEA-2019, and JFLEG.
- 3) Critical Analysis of Tradeoffs: To investigate the tradeoffs that are inherent in current GEC systems including that between inference latency (speed) and correction fluency (generative power) and to determine which architectures are most suitable when used in real-time processes as well as in offline processing.

- 4) Gap Identification: To determine enduring constraints in the field, including the so-called over-correction effect in LLMs and the language performance drop in low-resource languages, and therefore draw a clear roadmap to studies in the future.

III. RESEARCH METHOD

In order to present a strong comparative analysis of Grammatical Error Correction (GEC) systems, the research in the present study follows a systematic review model. This methodology of choice will make our statistical, neural, and transformer-based approach comparison unbiased, reproducible, and exhaustive. Instead of summarizing the existing literature, our methodology is aimed at tracking the technological path of the area, but it will process only studies that will provide measurable improvements to GEC architectures.

A. Research Objectives and Questions

The main objective of this review is to unravel the change of rigidity in the form of rules to neural fluidity in error correction. In order to direct this study, we established three narrow Research Questions (RQs) that address the architectural development, performance trade-offs, and performance standards of the profession:

TABLE I: Research Questions

No.	Research Question
RQ1	What has been the development history of grammatical error correction methods, and what distinguishes between them using statistical, neural and transformer-based methods?
RQ2	Which are the advantages and disadvantages of main GEC approaches in comparison and how do new transformer models do better?
RQ3	Which datasets and evaluation indicators are employed in research in the sphere of GEC, and what challenges and perspectives are in the way of the evolution of the field?

B. Data Sources and Search Strategy

our selected information retrieval strategy was aiming at locating high-impact studies, which were published over the past decade and a half (2013-2025). This period was carefully chosen to show the historical development of the field and reflect the thematic-chronological path of scientific progress of the field since the hallmark CoNLL- 2013 shared task to the latest developments presented by the power of Large Language Models.

We performed a searching action that concentrated on five leading academic repositories namely IEEE Xplore, ACL Anthology, ACM Digital library, Springer link, and Google scholar. The search terms were formulated as to intersect specific domain terms with architectural keywords:

- (Grammatical Error Correction AND (Deep Learning OR Transformer OR Seq2Seq OR Statistical Machine Translation) AND Grammatical Error Correction)

As a way of achieving maximum coverage, we used a snowballing method known as a backward snowballing (to identify foundational work that may have gone unnoticed during the initial keyword search), scrutinizing the reference lists of high-impact articles (e.g., the GECToR and T5 papers) to determine which works should be included in the search.

C. Selection Criteria

1) Inclusion Criteria:

- Content: Papers that propose a new algorithm, architecture or significant modification to existing GEC systems.
- Evaluation: Findings of the studies should be quantitatively reported in the form of standard scores on familiar data sets (CoNLL-2014, BEA-2019, JFLEG).
- Language & Type: peer-reviewed articles and conference papers written in English language only.

2) Exclusion Criteria:

- studies that were conducted on Error Detection only and did not involve a correction mechanism.
- Papers relying solely on private datasets that prevent reproducibility.
- theoretical papers that are not empirically validated.

IV. RELATED WORKS (LITERATURE REVIEW)

In this section, the application of Grammatical Error Correction (GEC) is discussed by reviewing key works in the area, and the ways the methods of the field changed since early statistical and rule-based methods to neural networks and ground-breaking transformers models. This would give a systematic knowledge of the changes of research directions with time, the advantages and limitations of each method, and the new models can give better results by the contextual learning and big scale pretrain. The literature is structured into three subsections, that is, Statistical & Rule-Based Approaches, Early Neural Architectures (RNNs & CNNs), and Transformers & Large Language Models (LLMs), to show the technological development and contrast the methods by a generation.

A. Statistical & Rule-Based Approaches

Formerly, GEC was established on a rule territory. The early systems consisted to a great extent of the linguistically founded principles and the statistical categorization of the mistakes. Neural methods are also important in critical uses where the domain is in which the issue of explainability is paramount, and in general-purpose contexts of unsupervised situations today where there is little labeled data available.

The drawbacks of this age could be best illustrated by means of the example of Sidorov et al. (2013) [11] who arrived at a rule-based approach towards CoNLL-2013 task. They relied on the NUCLE corpus and syntactic n-grams with dependency trees to come up with correct rules that govern such errors

as subject-verb agreement. Even though the authors thought that their F1-score of 3.3% is encouraging when the rules are manually constructed, the system was only able to recall 1.8% which demonstrates the basic flaw with the use of rules in data processing as compared to systems which use massive lexical representations. In order to overcome such a limitation of recalls, Rozovskaya and Roth (2016) [12] examined the synergistic performance of the Machine Learning Classifiers and Statistical Machine Translation (SMT). Their compilation of the flexibility of classifier (they used Averaged Perceptron and Naive Bayes) combined with the ability of the SMT to address complex interacting errors made them a hybrid pipeline and was trained on CoNLL-2014 and Lang-8. It received F0.5 of 47.40 that is greater than the state-of-the-art yet authors had to observe that SMT will require annotated data which is expensive and feature engineering is also expensive in classifiers.

Whereas the transfer has shifted to the neural models, the rule-based systems are still more efficient in the specialized world like in health field whereby the incidence of black box error is not an option. Gunter et al. (2022) [8] established the usefulness of the deterministic NLP algorithm called CHARTextract, which extracts information about stroke when reading radiology reports. Having a high accuracy of over 90% on specific occlusions, this paper showed that rule based systems are very specific, but with a weakness in the inconsistent reporting style. Similarly, Qian et al. (2025) [9] developed a RegEx and Boolean logic algorithm that can identify fall events in clinical notes with the sensitivity of 92.9 and specificity of 98.3. These articles uphold the fact that rule based systems continue to be instrumental in the proper extraction of data under controlled environments.

Statistical principles have of late been used in reviving unsupervised GEC in order to address the problem of data scarcity. Yasunaga et al. (2021) [13] proposed LM-Critic, a model that is trained on GPT-2 to make judgments about edits based on their statistical likelihood (perplexity), rather than being trained on edits. This system was competitive with other systems that were supervised but it happened to be highly costly at the time as it employed iterative scoring with CoNLL-14 having an F0.5 of 58.3. Equally in the example of low-resource language, Lin et al. (2023) [14] perceived GEC as a probabilistic scoring experiment with the Tagalog version of a BERT-based editor. Their experiment found them to be competitive in areas or contexts where Neural Machine Translation (NMT) traditional methods fail and they also claimed the pseudo-perplexity scoring was domain sensitive.

B. Early Neural Architectures (RNNs & CNNs)

Mapping of sequences (Seq2seq) was a breakthrough of mapping of the linguistic features to Deep Learning. It was the period that introduced the Encoder-Decoder paradigm that helped models to learn grammatical patterns directly working with the parallel corpora. These models were a transition phase to more liberal rules and modern Transformers that reflex contextualize them via Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), although lacked long-range interaction and vocabulary sizes.

The initial example of NMT to GEC presented by Yuan and Briscoe (2016) [15] set the move towards neural GEC. They used Encoder-Decoder based on RNN on the Cambridge Learner Corpus (CLC) and with attention on the corpus, and UNK tokens as an alignment in order to reach a new state of the art of F0.5 of 39.90% on CoNLL-2014. However, the model had a problem of huge vocabularies and non-public data. Chollampatt and Ng (2018) [16] proposed a Multilayer Convolutional Neural Network (CNN) as a way of overcoming the processing bottlenecks in RNNs that are sequential in nature. They utilized local context capture and efficiency in a better way with parallelizable convolutions and character n-grams (FastText), with an F0.5 of 54.79. It was however proposed by the study that it requires a heavy rescoring pipeline in order to achieve the best performance.

The next studies aimed at streamlining training and using data. Ge et al. (2018) [17] resolved the problem of generalization through the use of Bi-GRU model Fluency Boost Learning. They produced synthetic training pairs (back-boost, self-boost) to obtain an F0.5 of 54.51 but the multi-round inference made their computation more complex.

As Transformers started appearing, the RNNs remained optimized to perform a set of specific tasks since their footprint in resources is smaller. He (2021) [18] combined Word Embeddings with LSTM and attention to detect verb mistakes with an accuracy of 83.96, but recall was not as high due to the size of the dataset. Wang and Zhong (2022) [19] presented the ASS model that consists of the Seq2Seq RNNs, indicating the accuracy at 99.71 percent in detecting specific grammar, whereas Chen and Xiao (2024) [20] introduced a hybrid attention Bi-GRU model. These papers demonstrate that optimized RNNs can be used in resource-constrained settings with its accuracy versus efficiency despite being challenged with very long-range dependencies.

C. Transformers & Large Language Models (LLMs)

The current state of GEC is transformer-based architectures. These models have overcome the long-range dependencies issue of the RNNs with self-attention mechanisms and massive pre-training. This kind of research has been divided into two main directions, including Sequence-to-Edit (Seq2Edit) models (that pay emphasis on speed during inference) and Sequence-to-Sequence (Seq2seq) generative models (that pay emphasis on fluency and structure complexity).

1) The Efficiency and Edit-Tagging Paradigms,

Omelianchuk et al (2020) [5]. proposed the use of a Transformer-based model GECToR, which presents GEC as a sequence tagging model (insert, delete, replace). This technique allows the least alterations and has the capability to draw inferences within a short duration with rival F0.5. However, the authors have specified that it has been identified to have flaws with semantic restructuring in comparison to generative models. After this efficiency, Stahlberg and Kumar (2020) [21] proposed Seq2Edits, which predicts span-level operation of editing. It was an effective method of depending on good-quality edit annotations to eliminate unneeded rewriting though it was also quite effective.

2) Generative Models, LLMs, and Data Augmentation,

On the generative side, studies aim at determining the trade-off between fluency and precision. Making a step higher in the direction of multilingual generalization, Rothe et al. (2021) [22] turned mT5 into a text-to-text generative GEC system that accepted more than a hundred languages. Their methodology is based on contextual arguments and can provide fluent corrections based on rewriting an entire sentence rather than a line-by-line tagging. Evaluations of CoNLL-14, BEA-19, and non-English (German, Czech, and Russian) demonstrated significant progress, particularly in the low-resource conditions, provided with the assistance of transfer learning. The study, however, observed that the generative architecture becomes computationally inefficient at 11B parameter scale in large scale inference, although T5-style generation has been demonstrated to be effective with multilingual GEC. Katinskaia et al. (2024) [23] also tested the GPT-3.5, which, despite good recall, had hallucinations and over-correction.

To correct this over-correction and the lack of labeled data specifically, Zhao et al. (2019) [24] proposed a copy-augmented sequence-to-sequence Transformer. The structure of GEC tasks allows identifying that the majority of the tokens do not change, which means that the decoder directly copies the tokens in the source and preserves the meaning and works well with out-of-vocabulary words. Their method of pre-training denoising auto-encoders on the One Billion Word Benchmark and using multi-task targets at token and sentence levels resulted in radical accuracy and recall improvements on CoNLL-2014 and JFLEG. The contribution to the contemporary pipelines is a significant step that the unlabeled pieces of data and copying mechanisms are applied to advance fluency without interfering with fidelity.

To complete this data-based strategy, Yang et al. (2021) [25] developed a data augmentation system (Grammatical Error Generation) based on back-translation that was automated. They attained an F0.5 of 64.3 percent on CoNLL-2014 through repeated training but the amount of synthetic data generation was limited by computing resources.

3) Structural, Syntactic and Contextual Enrichments,

The recent attempts have been aimed at introducing linguistic structure to neural black boxes to introduce accuracy. The SynGEC by Zhang et al. (2022) [26] is a combination of a GEC-oriented dependency parser and Graph Convolutional Networks (GCN) into Transformer, and F0.5 of 68.4 on BEA-19. Similarly Cao et al. (2025) [3] also introduced CxGGEC based on CxG to identify more complex patterns even though it was incurring inferences of constructions and with an F0.5 of 73.5, but tokenization of constructions increased inference time. In their work, Liu et al. (2025) [27] applied the differentiable version of paragraph-wise fusion correction with BERT on phrases beyond the sentence level. They used syntactic correlations between the sentences, and obtained an F1 of 0.87 on FCE, but they stated that further effort is needed

to address cohesive and pragmatic features.

- 4) **Optimization Strategies and Adaptable Modules.** In order to advance the behavior of the models even further, scientists have developed advanced training schemes. The multi-task learning model suggested by Bout et al. (2023) [28] employed schedules optimized on the model and the authors demonstrated that the model can establish higher rates of convergence when compared to baseline models. Cao et al. (2023) [29] put forth a Hybrid Autoregressive (AR) and Non-Autoregressive (NAR) model to refine the predictions of low confidence, where F0.5 stood at 54.11, at the expense of the training. Transformers with GANs were utilized by Qin (2022) [30] to train the adversarial models that failed to identify semantic nuances and achieved a higher score in GLEU. Liang et al. (2025) [31] introduced Edit-Wise Preference Optimization, which is a model output fitting to human preference and reduces over-correction of Seq2Seq models. Finally, there are some specific modules which work on a single weakness: Hu et al. (2021) [32] applied BERT to correct misspellings in noisy text, and Taslimipour et al. (2022) [33] employed language features and Transformers to tackle Multi-word Expressions (MWEs) and significantly increased accuracy where general models have reduced performance.

V. MODELING DEVELOPMENT AND TRAINING

Grammatical Error Correction (GEC) model development represents the general direction of Natural Language Processing developing beyond deterministic symbolic systems into neural systems and ultimately to Transformer-based and large-scale generative models. This part outlines the major GEC modeling methods, which have been clarified on the structure of the design and regime of training, where special focus has been made on the current structural architecture and optimization strategies adopted across the field.

A. Modeling Approaches

Formerly, GEC was established on a rule territory. The early systems consisted to a great extent of the linguistically founded principles and the statistical categorization of the mistakes. Neural methods are also important in critical uses where the domain is in which the issue of explainability is paramount, and in general-purpose contexts of unsupervised situations today where there is little labeled data available.

- 1) **Statistical and Rule-Based Modeling Approaches** In older generations of grammatical error correction (GEC) systems, the explicit grammatical rules, handcrafted features and statistical estimates based on n-gram patterns or smoothed machine-translation based correction pipelines were heavily relied on. The models generally incorporated the following features:

- 1) linguistic preprocessing (POS tagging, parsing),
- 2) rule matching or template-driven correction,
- 3) statistical classifiers of the error types.

Although these methods had the benefit of interpretability and ability to control features, it had limitations due to

poor recall, long feature engineering, and inability to capture long-range linguistic dependencies.

- 2) **Neural Architectures: RNNs and CNNs** With the progress of neural sequence-to-sequence modeling, systems were also able to learn patterns of corrections directly provided by data. The RNN based architectures, like the LSTMs and GRUs, used contextual encoding of sentences, but the CNN encoders achieved efficiency by a parallel convolution. These models minimized a dependence on rules and handcrafted rules and facilitated:
- 1) contextual embeddings,
 - 2) supervised end-to-end learning
 - 3) data augmentation to improve robustness.

Despite these positive attributes, classical neural models could not cope with longer-range dependencies and vocabulary scaling, which inspired the use of Transformer architecture.

- 3) **Transformer-based and LLM-Based Approaches** Transformers also brought in self-attention techniques that absorb a global context enabling more profound semantic insight and pretraining at scale. There are two paradigms of Transformer of contemporary GEC studies:
- a) Edit-Based Transformer Models systems like GECToR and Seq2Edits take GEC as a sequence-tagging task, where endogenous edits (delete, replace, insert) occur. These models focus on
 - 1) little text modification,
 - 2) quick inference that can be used in real-time,
 - 3) how the error steps can be explained.
 - b) Generative Transformer Models and LLMs models like T5, multilingual mT5, copy-augmented Transformers and GPT-style large language models are used to rewrite sentences in entirety. They have strengths such as
 - 1) great fluency production,
 - 2) solid multilingual adaptation,
 - 3) adaptable syntactical structure reconstruction.

Very recent advances build upon these with syntax-enhanced models (e.g. SynGEC), construction-guided models (e.g. CxGGEC) and preference-optimization mechanisms that adapt the model output to accommodate more closely the human behaviour of correction. As detailed in Table 2, the transition from statistical classifiers to RNN/CNN seq2seq models and finally to Transformer-based architectures reflects the progressive shift toward contextualized representations and large-scale pretraining.

TABLE II: Comparison of GEC Approach Areas, Methods, and Representative Studies

Area	Method Categories	Studies
Statistical & Rule-Based	<ul style="list-style-type: none"> • Rule-based systems • Statistical Machine Translation (SMT) • Hybrid classifier models • Perplexity-based scoring GEC 	[11], [12], [13], [14]
RNN/CNN Era	<ul style="list-style-type: none"> • Seq2Seq RNN GEC • Attention GRU/LSTM • CNN Encoder-Decoder • RNN/CNN hybrid models 	[15], [16], [17], [18], [19], [20]
Transformers & LLMs	<ul style="list-style-type: none"> • Edit-based Transformers (Seq2Edit, GECToR) • Seq2Seq Transformer generation • Copy-augmented Transformers • Syntax-enhanced models (SynGEC, DepGEC) • Construction-guided models (CoGEC) • Multi-task training Transformers • LLM-based GEC (GPT-3.5) • GAN-based GEC • Preference optimization methods 	[21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]

B. Training Strategies

The studies of Grammatical Error Correction (GEC) models are usually based on the massive number of parallel corpora with erroneous sentences and their correct versions. In supervised fine-tuning, models are trained to minimize token-level or edit-level prediction losses with a cross-entropy loss function, which is frequently used in combination with label smoothing and teacher forcing to stabilize the training. Corpora like CoNLL-2014, BEA-2019, FCE, Lang-8, and W&I+LOCNESS which are frequently used in transformer and neural GEC experiments [5], [21], [22], [24], [28] offer heterogeneous errors and contexts using writing, so that models can extrapolate on top of small grammatical regularities. Parallel data is hence the mainstay of majority of modern systems particularly to edit-based Transformers and neural sequence-to-sequence models.

The role of pretraining in modern GEC modeling takes a center stage in modern GEC modeling. Transformer backbones others such as BERT, RoBERTa, and T5 are initially trained directly on large scale unlabeled corpora to learn about the linguistic, semantic and syntactic representations. These

pretrained features are then fine-tuned on GEC datasets and afterwards transferred to the task of grammatical correction. Research that uses this two-stage pipeline, including mT5-based multilingual GEC [22], GPT-3.5 adaptation to GEC [23], syntax-aware transformer of SynGEC [26], or hybrid autoregressive/non-autoregressive training [29], has shown significant improvement in cross-language and domain performance.

Since the GEC data is manually annotated versus other NLP areas, the use of data augmentation is highly prevalent to increase the area of training. Common strategies include synthetic error generation, back-translation and fluency-boost sampling, such as Fluency-Boost training [17] and multilingual T5 augmentation pipelines [22]. Other augmentation paradigms such as construction masking in CxGEC [3] or denoising pretraining with copy-augmented objectives in Zhao et al. [24] assist models to learn non-local syntax dependencies and phrase-level error patterns, which are present in actual data at a low rate.

New literature also puts focus on optimization-based training in which efforts aim at stabilizing learning and perfecting correction behavior. Multi-task learning schemes, like the ones by Bout et al. [28], combine training on auxiliary tasks (e.g. error detection, confidence scoring) to enhance precision and less wasteful editing. Fluency versus efficiency Balanced Hybrid decoding schemes, such as those that combine autoregressive and non-autoregressive predictions, like the one in Cao et al. [29], represent the balance between fluency and efficiency. The GAN-based GEC-related adversarial training [30] promotes more realistic correctional tendencies, whereas the edit-wise preference optimization [31] enhances the system behavior against human corrective tendencies and over-correction. Combined, these training strategies have become a key to the attainment of the competitive performance in the new GEC systems.

VI. DATASETS AND BENCHMARKING

The papers that were chosen in this review utilize diverse benchmark datasets, which have been selected in order to serve the purposes of a particular model. The use of datasets in each study individually makes it possible to see how the selection of dataset affects model performance, generalization, and coverage of errors.

Starting with rule-based and early statistical methods Sidorov et al.(2013) [11] were using the corpus of the CoNLL shared task, the NUCLE corpus. This data is highly edited and is specifically about certain types of grammatical errors, which is compatible with their rule-driven syntactic n-gram approach although it is less comprehensive than other large learner corpora. Conversely, Rozovskaya and Roth (2016) [12] increased the data they used by merging CoNLL-2014 training data, Lang-8 and Web1T native corpus. This combination of learner and native data enabled their hybrid SMT-classifier pipeline to correct more grammatical errors than Sidorov et al., who used only one dataset.

Yuan and Briscoe (2016) [15] used the Cambridge Learner Corpus (CLC) to train and tested their model on FCE and

CoNLL-2014, among the first neural models. In comparison to Sidorov et al. and Rozovskaya & Roth, who used smaller or mixed datasets, Yuan and Briscoe used a significantly larger and more detailed learner corpus, so that their NMT model could perform better than their predecessors in SMT systems. Chollampatt and Ng (2018) [16] employed both NUCLE and Lang-8 which provided them with much more training data than Yuan and Briscoe as Lang-8 does present large-scale sentences of learners even though the annotations are noisier. They tested CoNLL-2014 and JFLEG and hence were able to compare grammatical accuracy and fluency. Ge et al.(2018) [17] expanded the dataset space even more by incorporating Lang-8, CLC, NUCLE, and even synthetic data that was generated using fluency-boost measures. They therefore trained their model with the most diverse data combination of the neural systems in this group.

Other neural methods utilized more narrow datasets. He (2021) [18] also took a small subset of the CLEC corpus, with restricted the model to verb-related errors and restricted generalization. Wang and Zhong (2022) [19] applied a dataset on an English grammar evaluation task, which is on articles, nouns, verbs and prepositions. The model was very accurate as their data was more structured and smaller but specialized to particular grammar features. Chen and Xiao (2024) [20] employed the term standard English learner corpora and the trend of using learner-oriented datasets is maintained, though not based on the exceedingly large sets of the Lang-8-based researches.

The studies based on a transformer are based on wider and more varied data sets. Zhao et al.(2019) [24] trained their model with One Billion Word Benchmark and fine-tuned on ConLL-2014 and JFLEG, thus theirs is one of the only models that uses native corpus with learner data. Omelianchuk et al.(2020) [5] (GECToR) were trained on Lang-8, W&I + LOCNESS and NUCLE and tested on CoNLL-2014 and BEA-2019. Their dataset design is similar to the one suggested by Ge et al. but it contains current ERRANT aligned corpora. Rothe et al.(2021) [22] (mT5) went a step further to evaluate across CoNLL-2014, BEA-2019, and several non-English datasets and as such, theirs was the first multilingual system in the list of papers reviewed. The model of multi-task trained by Bout et al.(2023) [28] required W&I + LOCNESS to be trained and tested on CoNLL-2014 and BEA-2019, with their dataset selection options the most similar to GECToR but with additional task supervision.

A number of papers present more specialized datasets of grammatical reasoning. Qin (2022) [30] was also trained on Lang-8, CLC FCE, and NUCLE and used large noisy data along with high-quality learner corpora. CoLA, Lang-8 and FCE were employed by Liu et al.(2025) [27], which stressed the idea of the syntactic relations between the sentences, instead of focusing on the errors at the token level. Cao et al.(2023) [29] (Hybrid AR -NAR) has utilized huge learner corpus of FCE, Lang-8, NUCLE and W&I + LOCNESS, among other corpus of Chinese language, making their training systems one of the most multilingual. Zhang et al.(2022) [26] (SynGEC) trained CLang8, FCE, NUCLE, W&I + LOCNESS, and tested on CoNLL-2014 and BEA-2019, pointing at the

TABLE III: Datasets Used Across GEC Studies and Their Components

Dataset / Corpus	Type	Component / Description	Language(s)	Studies (Authors)
NUCLE	Dataset	Learner essays with manually annotated grammatical errors	English	[11], [12], [16], [17], [25], [5], [26], [3]
CoNLL-2014	Benchmark	ESL essays with gold grammatical corrections	English	[12], [15], [16], [17], [24], [30], [28], [25], [26], [3], [21], [31], [23]
FCE (First Certificate in English)	Benchmark	Cambridge learner essays with detailed grammatical annotations	English	[15], [27], [30], [29], [33], [26]
Lang-8	Dataset	Large crowd-sourced learner corpus with noisy parallel corrections	Multilingual	[12], [16], [17], [25], [5], [26], [3], [29], [27]
W&I + LOCNESS	Dataset / Benchmark	Learner and native essays annotated using ERRANT	English	[5], [28], [26], [3], [29], [23]
BEA-2019	Benchmark	Standard GEC benchmark using W&I + LOCNESS with official splits	English	[5], [26], [28], [3], [21], [31], [23]
JFLEG	Benchmark	Fluency-oriented sentences with multiple human rewrites	English	[16], [24], [30]
CLC (Cambridge Learner Corpus)	Dataset	Large learner corpus with detailed error types	English	[15], [17], [30], [33]
CLEC	Dataset	Chinese Learner English Corpus (verb-error subset)	English (Chinese learners)	[18]
One Billion Word	Dataset	Large native English corpus for LM pretraining and fluency modeling	English	[24], [25]
CLang8	Dataset	Cleaned and standardized Lang-8 subset	English	[5], [26], [3]
CoLA	Dataset	Corpus of Linguistic Acceptability	English	[27]
NLPCC-2018 GEC	Benchmark	Grammatical error correction dataset for Chinese	Chinese	[29]
COWS-L2H, RULEC-GEC, SweLL-gold	Benchmarks	Multilingual learner corpora (Germanic/Slavic languages)	Multilingual	[23]
Twitter Spelling Corpus	Dataset	Noisy misspellings and correction pairs from social media	English	[32]
MWE-Annotated FCE/CLC	Dataset	Multi-word-expression-focused learner corpora	English	[33], [2]

consistency of the dataset needed by syntax-enhanced architectures. The datasets of CLang8, W&I + LOCNESS, BEA and CoNLL-2014 selected by Cao et al.(2025) [3] (CxGEC) are almost identical to those used by SynGEC, with construction grammar markups added. Stahlberg and Kumar (2020) [21] tested Seq2Edits on BEA-2019 and CoNLL-2014, selecting the same benchmarks as GECToR and SynGEC, thus permitting direct comparisons between the edit based models.

Liang et al.(2025) [31] employed CoNLL-2014 and BEA-2019 to optimize editing preferences, which is the same evalua-

tion conditions of GECToR and SynGEC. Lastly, Katinskaia et al.(2024) [23] tested GPT-3.5 on CoNLL-2014, BEA-2019, FCE/W&I, COWS-L2H, RULEC-GEC, and SweLL-gold, the largest amount of all papers, and the only ones to use more than one language and learner group.

In general, the data choice of every paper is consistent with the modeling objectives. Models that rely on accurate grammatical boundaries also use curated corpora like NUCLE, FCE, and W&I + LOCNESS, whereas those that are interested in fluency or generalization use a wider range of resources,

including Lang-8, synthetic corpora, or large native languages. Transformer-based and LLM-based models are more flexible across languages and writing styles than the previous neural or rule-based models because they are likely to use the biggest and most varied collections of data. Table 3 summarizes the key datasets and benchmarks used across the reviewed GEC studies, highlighting their characteristics.

VII. EVALUATION AND ANALYSIS

This segment is a critical analysis of the performance curves of Grammatical Error Correction systems. Within the scope of the review of the literature, we compare the relative effectiveness of statistical, neural, and transformer-based methods by examining the empirical findings presented in the reviewed literature. Three main dimensions are analyzed, namely quantitative performance on standard benchmarks, trade-off between the inference Latency and the quality of the correction and qualitative considerations of over-correction and low-resource adaptation.

A. Quantitative Performance

The historical development of GEC is best measured by the score of the $F_{0.5}$ where precision is given twice the weight when compared to the recall in order to ensure the tool retains the user confidence in its writing support capabilities. The historical evolution of the performance with CoNLL-2014 test set demonstrates the clear architectural leap.

B. Statistical to Neural Transition

High baseline was set by the hybrid SMT-classifier system by Rozovskaya and Roth [12] whose $F_{0.5} = 47.40\%$. Though it was strong in the time, it was outcompeted by the early neural structures. Nonetheless, this did not occur at once; original RNN-based systems, e.g., Yuan and Briscoe [15], performed worse at the beginning (39.90%), since vocabulary bottlenecks were present. The maturity of pre-transformer neural GEC came with the introduction of CNN-based encoders by Chollampatt and Ng [16] with a 54.79% and showing that local context modeling, under the right conditions, could be a better model than pure statistical techniques.

C. The Transformer Supremacy

The most drastic performance improvements were brought by the introduction of Transformer-based architectures. The state-of-art models continuously achieve above the 60% mark on CoNLL-2014. It is worth noting that the combination of linguistic structures has been better than the black box end-to-end learning. As an example, Construction Grammar-guided model (CxGGEC) by Cao et al. [3] had an impressive score of $F_{0.5} = 73.5\%$ which was significantly higher than the baseline GECToR and T5 models. This indicates that large-scale pretraining is necessary, but syntactic relationship modeling (as found in SynGEC [26] that has 68.4% accuracy) provides the best precision.

D. Architectural Trade-offs: Generative vs. Edit-Based

One key dichotomy that has been recently observed in the literature is the trade-off between the generative models of the form of Seq2Seq and the tagging models of the form of Seq2Edit.

- **Inference Latency:** Edit-based models, that is, GECToR [5], turn out to be more appropriate in real-time systems. GECToR also does not employ the computational intensity of autoregressive decoding of T5 and GPT models to optimize the correction task to a sequence labeling problem (Insert, Delete, Replace).
- **Correction Power:** Generative models (e.g., T5, GPT-3.5) on the other hand possess more correction power on more complex fluency errors. Unlike GECToR, which limits itself to set preset tags, generative techniques can rephrase entire clauses to create coherence. However, this power includes a cost of semantic drift and hallucination, as Katinskaia et al. [23] note in the case of GPT-3.5, this power is also accompanied by the cost of changing the original intent of the user.

E. Generalization and Resource Constraints

The analysis indicates that there is a long-term gap in terms of resources available in both the resource-rich and low-resource environments.

1) *The Over-Correction Phenomenon:* Large Language Models (LLMs), are inclined to over-correction. Compared with statistical models that had low recall (e.g., Sidorov et al. [11] had a recall of 3.3%), LLMs are likely to propose stylistic edits which are not necessarily grammatical mistakes. Liang et al. [31] have taken this up through Edit-Wise Preference Optimization which proved that alignment with human preferences has become no less important than raw accuracy.

2) *Multilingual Scalability:* Whereas English GEC has achieved a certain degree of maturity where studies are done on marginal gains in $F_{0.5}$, multilingual GEC is still at the developmental phase. Rothe et al. [22] have shown that massive multilingual models (mT5) can be used to do zero-shot transfer, but are less accurate than English-specific models. According to the analysis, the further enhancement of non-English GEC is expected to be based on the synthetic data production (as suggested by Yang et al. [25]) instead of the costly production of annotated corpora such as FCE or BEA-2019.

F. Summary of Findings

The comparison analysis results in three conclusions:

- 1) **Structure Matters: Data-based Transformers:** Hybrid systems that use syntactic priors (dependency trees, construction grammar) are taking over the lead over pure data-driven Transformers.
- 2) **Speed-Quality Pareto Frontier:** The current is that there is no one-size-fits-all, but Seq2Edit is required when the application is latency-sensitive, and Seq2Seq is required when the application requires the high fluency editing.
- 3) **Data Dependence:** It is becoming increasingly typical that the performance improvements are not coming solely

through architectural novelty but because of data augmentation and artificial error generation.

VIII. CONCLUSION

This work has given a detailed comparative analysis of Grammatical Error Correction (GEC) methods and how they have evolved since their simple statistical algorithmic approaches to the more advanced transformer based designs. We can refer to the three research questions developed in Section III after conducting a systematic analysis of the recent literature (2013-2025) and assessing the performance measures in terms of the key benchmarks. Having made the comprehensive comparison as it has been addressed in the earlier sections, as we conclude our work, we enumerate our key findings according to each RQ.

A. (RQ1) What has been the development history of grammatical error correction methods, and what distinguishes between them using statistical, neural and transformer-based methods?

It is plausible to assume that the field of deep reinforcement combined with reinforcement learning would transform this into a model that produces neural-equivalent manipulation of the context.;—human—;1) THE RIGID-rule to Context-Aware Transformer Trajectory It is reasonable to believe that the field of deep reinforcement with reinforcement learning would convert this into a model that generates neural-equivalent manipulation of the context.

- GEC history can be described as a move in the direction of the strict prescription of language to the generation of data.
- Statistical Era: These systems were based on rules and SMT that were handcrafted giving them high interpretability but poor recall. They were unable to get mistakes that are based on long-range context.
- Neural Era: Characterized by the introduction of RNNs as well as CNNs, this was the start of what is termed end-to-end learning, where feature engineering is no longer necessary. The bottlenecks of these models, however, were the size of vocabulary and parallelization of training.
- Transformer Era: The state-of-the-art that is presently available stands out as a Self-Attention mechanism. Transformer models (e.g., BERT, T5) unlike their predecessors operate on the whole input sequence at once, which enables them to correct complex, long-range dependencies that previous models missed systematically.

B. (RQ2) Which are the advantages and disadvantages of main GEC approaches in comparison and how do new transformer models do better?

Transformer-based models are far superior to statistical and early neural models, we find, and they are guaranteed to achieve more than 60% in F0.5 in CoNLL-2014. However, this approach of the modern approach has a clear dichotomy:

- Seq2Edit (e.g., GECToR): This primarily is inference speed. Such models suit the time-related applications

but are limited when the restructuring of the complex sentences is taken into consideration as GEC is a tagging task (Insert/ Delete /Replace).

- Seq2Seq (e.g., T5, GPT): These have the advantage of being able to generate and to be fluent. They can also write down all the paragraphs afresh so as to make them coherent. The disadvantage, however, lies in the fact that it can also over-correct other styles, the intention of the author in the style can be distorted, and the computation costs are also higher.

C. (RQ3) Which datasets and evaluation indicators are employed in research in the sphere of GEC, and what challenges and perspectives are in the way of the evolution of the field?

- 1) Standardization of benchmarking and the problem of multilingual scalability.

The field has been normalized on major standards of English: CoNLL-2014 to grammatical accuracy and BEA-2019/JFLEG to fluency. Although this standardization has provided rigorous comparison, it has created an imbalance on the resources.

- The Data Bottleneck: The application of fully supervised data (parallel corpora) is also a massive challenge. Our review has demonstrated that the way forward is synthetic data generation and semi-supervised learning to deal with the unavailability of annotated data.
- Low-Resource Languages: Most of the developments are Anglicentric. These large-scale models will be altered in the next decade to low-resource languages without the prohibitive cost of creating large labeled datasets.
- Future Direction: We believe that the future of GEC is not only bigger models, but hybrid architectures with the generative ability of LLMs and the accuracy of linguistic structures (e.g., Syntax-Enhanced Transformers), such that can provide a correct answer and also reflect the original meaning.

D. Limitations of this review

There are some weaknesses to this comparative review worth noting. Firstly, we only selected studies that were published in English and this could have left out other GEC developments in other languages. Second, we focused on high-impact papers of major NLP conferences (ACL, EMNLP, IEEE) but due to the fast pace of development of the art, some pre-prints or very recent proprietary models are not always exhaustively covered. Lastly, the direct comparison of F0.5 scores between studies is occasionally complicated by the dissimilarity in the evaluation protocols of the initial articles.

Overall, this comparative review demonstrates that despite impressive achievements of Grammatical Error Correction with Transformer-based models, the path of the uncomplicated error correction to the multifunctional writing support is not completed yet. The intersection between language structure

and the abilities of Large Language Models to generate is the future of the area. In solving the constraints of the present in over-correction and adjustability, future studies will open the way to powerful, situation-sensitive, and intuitive smart writing helpers to serve a worldwide customer base.

IX. FUTURE WORK

Future research directions in grammatical error correction could concentrate on domain adaptation, as general-purpose models are known to lack performance when working with data from specialized domains. Another area of study should be low-resource languages, for which multilingual and unsupervised approaches are proving promising but are still limited in capturing complex grammatical structure. Rare error types, especially in multiword expressions and idioms, which benefit from correction but are being frequently miscorrected in current models, also stand as an important area of investigation. Equally important would be a focus on issues related to bias, fairness, and overcorrection, where large generative models like GPT-3.5 are seen to rewrite when it's not needed and affect meaning. Another direction would relate to efficiency, where many of the current successful models in this area are computationally expensive. Another area which would emerge due to increasing use of LLMs would relate to their increasing incorporation with current Grammatical Error Correction models, possibly in re-ranking, evaluation, and hybrid models.

ACKNOWLEDGMENTS

The authors would wish to personally thank Imam Abdulrahman Bin Faisal University and the College of Computer Science and Information Technology not only in availing the means and the support that they required to complete this study. We also wish to thank the Department of Computer Engineering whose guidance in academics helped us. We want to particularly extend our gratitude to our supervisor, Dr. Mostafa Youldash, whose mentorship and support in this project were invaluable and were always present.

APPENDIX

APPENDIX A: SUMMARY OF REVIEWED STUDIES IN RELATED WORKS (LITERATURE REVIEW)

TABLE IV: Data Extraction Table

Ref.	Author(s)	Year	Title	Dataset	ML/DL (model)	Results
[3]	Cao et al.	2025	CxGEC: Construction-Guided Grammatical Error Correction	English: Lang-8; Chinese: NLPCC/GEC datasets	Seq2Seq GEC + CxG patterns	Outperforms baseline seq2seq on English/Chinese benchmarks.
[5]	Omelianchuk et al.	2020	GECToR – Grammatical Error Correction: Tag, Not Rewrite	BEA-2019, CoNLL-2014	Transformer Seq2Edits	State-of-the-art F0.5; faster inference.
[8]	Gunter et al.	2022	Rule-based NLP for automation of stroke data extraction	773 Radiology reports	Rule-based NLP (CHARTextract)	~90% accuracy distal/posterior occlusion; 85% proximal occlusion.
[9]	Qian et al.	2025	Rule-Based NLP to Identify Falls in Older Adult Inpatient Records	Inpatient records (Hong Kong), n=1000	Rule-based NLP	Sensitivity 93.3%, Specificity 99.0%, F1 0.903.
[11]	Sidorov et al.	2013	Grammar Correction Using Syntactic N-grams	NUCLE corpus (CoNLL-2013)	Rule-based system	Precision 17.4%, Recall 1.8%; baseline system.
[12]	Rozovskaya et al.	2016	GEC: Machine Translation and Classifiers	CoNLL-2014, Lang-8, Web1T	Classifiers + SMT (Moses)	F0.5 47.40 (+20% relative improvement).
[13]	Yasunaga et al.	2021	LM-Critic: Language Models for Unsupervised GEC	CoNLL-2014, BEA-2019, GMEG-wiki/yahoo	BIFI + LM-Critic (GPT-2)	Unsupervised: +7.7 avg F0.5; Supervised: F0.5 65.8–72.9.
[14]	Lin et al.	2023	BERT-based Unsupervised GEC	Tagalog, Indonesian GEC corpus	BERT (multi-order pseudo-perplexity)	Tagalog F0.5 0.6932 vs 0.3618; Indonesian F0.5 0.6518 vs 0.5510.
[15]	Yuan et al.	2016	GEC using Neural Machine Translation	FCE, CoNLL-2014	NMT (RNN Encoder-Decoder + attention)	First NMT to beat SMT; F0.5 39.90.
[16]	Chollampatt et al.	2018	Multilayer Convolutional Encoder-Decoder for GEC	Lang-8, NUCLE, CoNLL-2014, JFLEG	CNN Encoder-Decoder	F0.5 54.79, GLEU 57.47.
[17]	Ge et al.	2018	Fluency Boost Learning and Inference for Neural GEC	Lang-8, CLC, NUCLE, CoNLL-14, JFLEG	Seq2Seq Bi-GRU + Fluency Boost	F0.5 54.51, GLEU 57.74.
[18]	He	2021	English Grammar Error Detection Using RNNs	CLEC	Bi-LSTM + Transformer Attention	Precision 0.8396, Recall 0.2965, F1 0.7725; verb F1 0.3599.
[19]	Wang et al.	2022	English Grammar Check	GEC test sets	ASS Model / Seq2Seq	Test set: Acc 99.71%, F1 98.82%.
[20]	Chen et al.	2024	Intelligent Error Correction with Hybrid Attention	CoNLL-2014, JFLEG, CLEC	Bi-GRU + Attention + Transformer	F0.5 60.74%, GLUE 61.13.
[21]	Stahlberg et al.	2020	Seq2Edits: Sequence Transduction Using Span-Level Edit Ops	BEA-2019, CoNLL-2014	Edit-based Seq2Seq	Efficient editing; strong GEC performance.
[22]	Rothe et al.	2021	Multilingual GEC Recipe	BEA-19, CoNLL-14, RULEC, Falko-MERLIN, AKCES	mT5 text-to-text model	Strong multilingual performance.
[23]	Katinskaia et al.	2024	GPT-3.5 for GEC	Multiple languages	GPT-3.5 (LLM)	Zero-shot high recall; best with re-ranking.
[24]	Zhao et al.	2019	Improving GEC via Copy-Augmented Seq2Seq	CoNLL-2014, JFLEG	Copy-augmented Seq2Seq	Large performance boost; better copy retention.
[25]	Yang et al.	2021	NMT for Detecting and Correcting Grammar Errors	NUCLE, FCE, W&I-LOCNESS, Lang-8	Transformer NMT	F0.5 64.3%; surpasses baselines.
[26]	Zhang et al.	2022	SynGEC: Syntax-Enhanced GEC	CoNLL-14, BEA-19, NLPCC-18, MuCGEC	Transformer + DepGCN	Consistent improvements.
[27]	Liu et al.	2025	Paragraph Grammar Correction via Differential Fusion	COLA, LCOLE, FCE	BERT + Transformer	High accuracy/F1.
[28]	Bout et al.	2023	Efficient GEC via Multi-Task Training	BEA-19, CoNLL-14	Transformer (BART) + Multi-Task	Enhanced F0.5; reduced computation.
[29]	Cao et al.	2023	Improving Autoregressive GEC with Non-autoregressive Models	BEA-19, CoNLL-14, JFLEG	Autoregressive Transformer	Improved F0.5 across datasets.
[30]	Qin	2022	Automatic Correction via Deep Learning	Training: Lang-8, CLC FCE, NUCLE; Test: CoNLL-2014, JFLEG	Transformer + GAN	F0.5 53.87; GLEU 61.77.
[31]	Liang et al.	2025	Edit-Wise Preference Optimization	CoNLL-2014, BEA-2019	Transformer Seq2Edit + Preference	Reduces over-correction; improves precision.
[32]	Hu et al.	2021	Misspelling Correction with Pre-trained LM	Twitter dataset	BERT + edit-distance	Better context-sensitive spelling correction.
[33]	Taslimipoor et al.	2022	Improving GEC for Multiword Expressions	Cambridge Learner Corpus, FCE	Transformer + MWE-aware	Improved correction of MWEs; higher phrase-level precision.

APPENDIX B: SKILLS LEARNED

The experience of working on this comparative review has been a significant learning experience to our entire team. We developed a strong understanding of the concepts of Natural Language Processing (NLP), namely, the architectural development of the Statistical and Rule-based systems to the more sophisticated Transformer-based models. We learned much about trade-offs between Sequence-to-Edit models (such as GECToR) to be efficient and Sequence-to-Sequence generative models (such as T5) to be fluent. We learned to apply a systematic review to understand the historical development of GEC between 2013 and 2025, as well as identify the most important research gaps, including the over-correction phenomenon in LLMs and the performance problems in low-resource languages. And finally, we were taught how to write our scholarly discoveries in IEEE format and how to be good team players and project managers so as to meet our submission deadline.

REFERENCES

- [1] A. Masciolini, A. Caines, O. De Clercq, J. Kruijsbergen, M. Kurfahl, R. Muñoz Sánchez, E. Volodina, R. Östling, K. Allkivi, S. Arhar Holdt *et al.*, “An overview of grammatical error correction for the twelve multigec-2025 languages,” 2025.
- [2] C. Bryant, Z. Yuan, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, “Grammatical error correction: A survey of the state of the art,” *Computational Linguistics*, vol. 49, no. 3, pp. 643–701, 2023.
- [3] Y. Cao, T. Wang, L. Xu, Z. Wang, and M. Cai, “Cxgtec: Construction-guided grammatical error correction,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 6143–6156.
- [4] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, “Recent advances in natural language processing via large pre-trained language models: A survey,” *ACM Computing Surveys*, vol. 56, no. 2, pp. 1–40, 2023.
- [5] K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. Skurzhanskyi, “Gector-grammatical error correction: tag, not rewrite,” *arXiv preprint arXiv:2005.12592*, 2020.
- [6] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, “A survey of large language models,” *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.
- [7] R. He, J. Cao, and T. Tan, “Generative artificial intelligence: a historical perspective,” *National Science Review*, vol. 12, no. 5, p. nwaf050, 2025.
- [8] D. Gunter, P. Puac-Polanco, O. Miguel, R. E. Thornhill, A. Y. Yu, Z. A. Liu, M. Mamdani, C. Pou-Prom, and R. I. Aviv, “Rule-based natural language processing for automation of stroke data extraction: a validation study,” *Neuroradiology*, vol. 64, no. 12, pp. 2357–2362, 2022.
- [9] X. X. Qian, P. H. Chau, D. Y. Fong, M. Ho, and J. Woo, “Development and validation of a rule-based natural language processing algorithm to identify falls in inpatient records of older adults: Retrospective analysis,” *JMIR aging*, vol. 8, p. e65195, 2025.
- [10] M. K. Keita, C. Homan, M. Zampieri, A. Bremang, H. A. Alfari, E. A. Ibrahim, and D. Owusu, “Grammatical error correction for low-resource languages: The case of zarma,” *arXiv preprint arXiv:2410.15539*, 2024.
- [11] G. Sidorov, A. Gupta, M. Tozer, D. Catala, A. Catena, and S. Fuentes, “Rule-based system for automatic grammar correction using syntactic n-grams for english language learning (l2),” in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, 2013, pp. 96–101.
- [12] A. Rozovskaya and D. Roth, “Grammatical error correction: Machine translation and classifiers,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2205–2215.
- [13] M. Yasunaga, J. Leskovec, and P. Liang, “Lm-critic: Language models for unsupervised grammatical error correction,” *arXiv preprint arXiv:2109.06822*, 2021.
- [14] N. Lin, H. Zhang, M. Shen, Y. Wang, S. Jiang, and A. Yang, “A bert-based unsupervised grammatical error correction framework,” *arXiv preprint arXiv:2303.17367*, 2023.
- [15] Z. Yuan and T. Briscoe, “Grammatical error correction using neural machine translation,” in *Proceedings of the 2016 conference of the north American Chapter of the Association for computational linguistics: Human language technologies*, 2016, pp. 380–386.
- [16] S. Chollampatt and H. T. Ng, “A multilayer convolutional encoder-decoder neural network for grammatical error correction,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [17] T. Ge, F. Wei, and M. Zhou, “Fluency boost learning and inference for neural grammatical error correction,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1055–1065.
- [18] Z. He, “English grammar error detection using recurrent neural networks,” *Scientific Programming*, vol. 2021, no. 1, p. 7058723, 2021.
- [19] X. Wang and W. Zhong, “Research and implementation of english grammar check and error correction based on deep learning,” *Scientific Programming*, vol. 2022, no. 1, p. 4082082, 2022.
- [20] S. Chen and Y. Xiao, “An intelligent error correction model for english grammar with hybrid attention mechanism and rnn algorithm,” *Journal of Intelligent Systems*, vol. 33, no. 1, p. 20230170, 2024.
- [21] F. Stahlberg and S. Kumar, “Seq2edit: Sequence transduction using span-level edit operations,” *arXiv preprint arXiv:2009.11136*, 2020.
- [22] S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn, “A simple recipe for multilingual grammatical error correction,” *arXiv preprint arXiv:2106.03830*, 2021.
- [23] A. Katinskaia and R. Yangarber, “Gpt-3.5 for grammatical error correction,” *arXiv preprint arXiv:2405.08469*, 2024.
- [24] W. Zhao, L. Wang, K. Shen, R. Jia, and J. Liu, “Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data,” *arXiv preprint arXiv:1903.00138*, 2019.
- [25] D. Yang, X. Sun, and P. Wang, “Using neural machine translation for detecting and correcting grammatical errors,” in *Proc. 20th Int. Conf. WWW/Internet Appl. Comput.*, 2021, pp. 11–18.
- [26] Y. Zhang, B. Zhang, Z. Li, Z. Bao, C. Li, and M. Zhang, “Syngec: Syntax-enhanced grammatical error correction with a tailored gec-oriented parser,” *arXiv preprint arXiv:2210.12484*, 2022.
- [27] W. Liu, C. Zhao, Y. Li, C. Cai, H. Liu, R. Qiu, R. Su, and B. Li, “A method for english paragraph grammar correction based on differential fusion of syntactic features,” *Plos one*, vol. 20, no. 7, p. e0326081, 2025.
- [28] A. Bout, A. Podolskiy, S. Nikolenko, and I. Piontkovskaya, “Efficient grammatical error correction via multi-task training and optimized training schedule,” *arXiv preprint arXiv:2311.11813*, 2023.
- [29] H. Cao, Z. Cao, C. Hu, B. Hou, T. Xiao, and J. Zhu, “Improving autoregressive grammatical error correction with non-autoregressive models,” in *Findings of the Association for Computational Linguistics: ACL 2023*, 2023, pp. 12014–12027.
- [30] M. Qin, “A study on automatic correction of english grammar errors based on deep learning,” *Journal of Intelligent Systems*, vol. 31, no. 1, pp. 672–680, 2022.
- [31] J. Liang, H. Yang, S. Gao, and X. Quan, “Edit-wise preference optimization for grammatical error correction,” in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 3401–3414.
- [32] Y. Hu, X. Jing, Y. Ko, and J. T. Rayz, “Misspelling correction with pre-trained contextual language model,” in *2020 ieee 19th international conference on cognitive informatics & cognitive computing (icci* cc)*. IEEE, 2020, pp. 144–149.
- [33] S. Taslimipoor, C. Bryant, and Z. Yuan, “Improving grammatical error correction for multiword expressions,” in *Proceedings of the 18th Workshop on Multiword Expressions@ LREC2022*, 2022, pp. 9–15.

BIOGRAPHY

Anwar Aldahan is pursuing the B.S. degree in Artificial Intelligence at the Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her interests include linguistically informed AI models and computational language engineering.

Ohoud Alqria is pursuing the B.S. degree in Artificial Intelligence at the Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interests include natural language processing and deep learning applications.

Fatimah Alwarsh is pursuing the B.S. degree in Artificial Intelligence at the Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her research interests include machine learning, computational linguistics, and intelligent language technologies.

Shahad Alshehab is pursuing the B.S. degree in Artificial Intelligence at the Department of Computer Engineering, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. Her interests include neural language models, NLP systems, and AI-driven educational technologies.