UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Enhancing Grammatical Correctness: The Efficacy of Large Language Models in Error Correction Task

Author:
Oleksandr KORNIIENKO

Supervisor:
Kostiantyn OMELIANCHUK

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2024

Declaration of Authorship

I, Oleksandr KORNIIENKO, declare that this thesis titled, “Enhancing Grammatical Correctness: The Efficacy of Large Language Models in Error Correction Task” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Enhancing Grammatical Correctness: The Efficacy of Large Language Models in Error Correction Task

by Oleksandr KORNIIENKO

Abstract

Recent studies have highlighted the exceptional capabilities of open-sourced foundational models like LLaMA, Mistral, and Gemma, particularly in scenarios requiring writing assistance. These models demonstrate proficiency in various tasks both in zero-shot settings and when fine-tuned with task-specific, instruction-driven data. Despite their adaptability, the application of these models to Grammatical Error Correction (GEC) tasks, critical for producing grammatically accurate text in writing assistants, remains underexplored. This thesis explores the performance of open-sourced Large Language Models (LLMs) in GEC task across multiple setups: zero-shot, supervised fine-tuning, and Reinforcement Learning from Human Feedback (RLHF). Our research shows that task-specific fine-tuning significantly enhances LLM performance on GEC tasks. We also highlight the importance of precise prompt configuration in zero-shot settings to align models with the specific requirement of the CoNLL-2014 and BEA-2019 benchmarks, aiming for minimal necessary edits. Further, our experiments with RLHF, particularly Direct Preference Optimization, provide insights into aligning LLMs for specific applications, showing an improvement of 0.3% in scores and indicating a further path for improvement. The best-performing model, Chat-LLaMA-2-13B-FT, matched the performance of state-of-the-art models with considerably less data, achieving an $\mathbf{F_{0.5}}$ score of 67.87% on the CoNLL-2014-test and 73.11% on the BEA-2019-test benchmarks. This thesis expands our understanding of the capabilities of open-sourced LLMs in GEC and sets the stage for future enhancements in this area. The code and trained model are publicly available.

Acknowledgements

I am grateful to my supervisor, Kostiantyn Omelianchuk, whose guidance was invaluable throughout my research journey. His support, patience, and insightful contributions were fundamental to the completion of this thesis.

I sincerely thank Grammarly for providing the essential computational resources and the team, especially Oleksandr Skurzhanskyi and Artem Chernodub, for their valuable feedback that enabled this research.

I am also grateful to the Ukrainian Catholic University for their outstanding master’s program. Special thanks to Oleksiy Molchanovsky, Daryna Petrenko, and Ruslan Partsey for their coordination and invaluable advice throughout my education.

During this challenging time when Ukraine continues to resist the unprovoked Russian invasion, I am thankful to our communities and to everyone who defends our homeland, supports our people, and sends aid amidst the ongoing conflict.

My deepest appreciation goes to my family and wife, whose unwavering support and love are my pillars of strength.

Finally, I would like to thank my group mates for their camaraderie and for all the memorable moments we shared, which made this journey enjoyable and enriching.

Contents

List of Figures

List of Tables

List of Abbreviations

BEA-2019	**B**uilding **E**ducational **A**pplications 2019 shared task
DPO	**D**irect **P**reference **O**ptimization
FN	**F**alse-**N**egative
FP	**F**alse-**P**ositive
FT	**F**ine-**T**uning
GEC	**G**rammatical **E**rror **C**orrection
LLM	**L**arge **L**anguage **M**odel
M2	**M**ax **M**atch score
NLP	**N**atural **L**anguage **P**rocessing
NMT	**N**eural **M**achine **T**ranslation
RL	**R**einforcement **L**earning
RLHF	**R**einforcement **L**earning from **H**uman **F**eedback
SFT	**S**upervised **F**ine-**T**uning
SOTA	**S**tate-**O**f-**T**he-**A**rt
TN	**T**rue-**N**egative
TP	**T**rue-**P**ositive
W&I	**W**rite **I**mprove dataset
ZS	**Z**ero-**S**hot

Chapter 1

Introduction

1.1 Motivation

Grammatical Error Correction (GEC) is a crucial component of Natural Language Processing (NLP) that significantly enhances text writing tasks, especially for non-native speakers, to produce complex texts and improve their writing performance and clarity, which is crucial in business communication. The development of GEC began in the 1980s (Kwasny and Sondheimer, 1981; Jensen et al., 1983) and continues to evolve today (Bryant et al., 2023).

GEC consists of correcting sporadic spelling, punctuation, and word choice mistakes but also involves enhancing text fluency and clarity with minimal edits, considering learner-specific suggestions, and understanding context (Napoles, Sakaguchi, and Tetreault, 2017). Consequently, researchers have shifted from traditional rule-based systems to data-driven sequence-to-sequence (seq2seq) approaches (Junczys-Dowmunt et al., 2018) that utilize deep learning and Neural Machine Translation (NMT). These approaches rely heavily on large training datasets (Tarnavskyi, Chernodub, and Omelianchuk, 2022; Rothe et al., 2021), making their development data resource-intensive, limiting scalability, particularly when adapting to specific tasks.

Recently, there has been rapid development in writing assistants powered by Large Language Models (LLMs) with capacities exceeding 1 billion parameters and general-purpose language generation capabilities (Radford et al., 2019). These models can handle various tasks (Brown et al., 2020), including GEC. Studies Loem et al., 2023; Fang et al., 2023 show that LLM-powered writing assistants like ChatGPT have excellent error detection capabilities and can correct errors to make sentences more fluent, often due to over-correction that does not align with the principle of minimal edits.

Recent research by Omelianchuk et al., 2024 in GEC using seq2seq models has highlighted the significant potential for improvements in these systems. Therefore, further research and development could considerably impact advancing the GEC industry.

We selected English GEC system development for our study because it has seen the most progress, with the most annotated training data and extensive research outcomes (Bryant et al., 2023). However, it is important to note that our approach, which is based on Large Language Models, supports multi-lingual tasks and can be applied to other languages. In our work, we focus on enhancing current seq2seq approaches using LLMs for English, leaving the development of systems for other languages to future research.

1.2 Goals of the master research

The goal of this Master's thesis is to investigate and enhance sequence-to-sequence approaches with the application of Large Language Models for Grammatical Error Correction. We aim to advance the performance on prominent benchmarks, specifically the CoNLL-2014 and BEA-2019 shared tasks. We set the following goals:

1. We would like to perform an analysis of existing GEC approaches and define the strengths and weaknesses of existing systems. This analysis will help us identify best practices and areas where improvements can be made, particularly in enhancing the quality of corrections through fine-tuning strategies.

2. Then, we will explore how different prompt formulations in a zero-shot setting affect the performance metrics of LLMs, with a particular focus on the precision of grammatical error corrections. This investigation will include experimenting with various prompt modifications to determine their influence on the model's ability to correct errors accurately.

3. We also plan to assess the effectiveness of supervised fine-tuning by examining various components, such as the choice of GEC datasets and model size. We also will compare full model fine-tuning against parameter-efficient tuning methods to determine the best practices for optimizing performance while managing computational resources efficiently.

4. We aim to integrate RLHF methods to refine LLM performance using human-like preference data on GEC task. This approach will be tested to see how well it enhances the model's ability to align with preference data, thus potentially improving the metrics on GEC benchmarks.

1.3 Structure of thesis

In Chapter 1, we provide an overview of the Grammatical Error Correction field and outline the research goals for this thesis. This initial chapter sets the stage for an exploration of the topic and establishes the framework for subsequent investigations.

Chapter 2 reviews existing GEC models, including sequence-to-sequence, sequence-to-tag, and ensembling methods. This review identifies gaps in the current research and helps formulate strategies for improvement, aiming to advance the state of the art in GEC methodologies.

Chapter 3 examines publicly available English datasets commonly used in GEC research. We also discuss the metrics employed to evaluate the performance of GEC models, providing a basis for consistent and reliable benchmarking across different studies.

Chapter 4 details our experiments within various training frameworks, including zero-shot inference settings and supervised fine-tuning. We also introduce Direct Preference Optimization as an alternative to traditional Supervised Fine-Tuning. This chapter aims to evaluate approaches considered to enhance the performance of GEC models, with a focus on how different training settings affect model quality.

Throughout these chapters, we aim to build a coherent narrative that highlights the technical aspects of GEC and contextualizes our research within the broader academic and practical realms of Natural Language Processing.

Chapter 2

Literature review

Grammatical Error Correction field commenced with rule-based methods in the early 1980s (Kwasny and Sondheimer, 1981; Jensen et al., 1983) and has since evolved into sophisticated data-driven approaches, employing supervised machine learning models trained on annotated corpora of error-laden text with exemplary suggestions. Pioneering investigations in this domain include the seminal works of (Brockett, Dolan, and Gamon, 2006; De Felice and Pulman, 2008; Rozovskaya and Roth, 2010; Tetreault, Foster, and Chodorow, 2010; Dahlmeier and Ng, 2012).

2.1 Existing GEC systems

A seminal moment in the analysis and correction of grammatical errors unfolded with the establishment of The Helping Our Own shared task in 2012, marking a significant shift from rule-based to data-driven methods and releasing the First Certificate in English (FCE) corpus in Yannakoudakis, Briscoe, and Medlock, 2011. Subsequently, Leacock et al., 2010 conducted a comprehensive survey summarizing the progress in the field.

Over the past decade, the evaluation of grammatical error correction systems has advanced significantly through key public benchmarks such as CoNLL-2013 (Ng et al., 2013), CoNLL-2014 (Ng et al., 2014), and BEA-2019 (Bryant et al., 2019). The $\mathbf{F_{0.5}}$ score, utilized in the MaxMatch (M2) (Dahlmeier and Ng, 2012) and ERRANT (Bryant, Felice, and Briscoe, 2017) metrics, has become the primary standard for evaluating GEC systems. Studies by Grundkiewicz, Junczys-Dowmunt, and Gillian, 2015 and Chollampatt and Ng, 2018 demonstrate that the $\mathbf{F_{0.5}}$ score aligns closely with human judgment, underscoring its effectiveness and reliability in assessing grammatical accuracy.

A notable breakthrough in GEC was the incorporation of deep learning methodologies, which are mostly used in Machine Translation tasks. The introduction of Transformer architectures in Vaswani et al., 2023; Junczys-Dowmunt et al., 2018, marked a significant evolution in the field. The application of these architectures has enhanced the performance of GEC systems and aligned them more closely with the latest advances in machine learning and natural language processing. This integration highlights the continuous evolution of GEC systems, leveraging cutting-edge technologies to improve their accuracy and efficiency.

Presently, two primary methodologies are employed for constructing GECs using Neural Machine Translation. The first set of methods leverages low-resource sequence-to-sequence approaches (Yuan, Briscoe, and Felice, 2016). These methods augment text generation by incorporating additional contextual information during encoding, such as BERT representations (Kaneko et al., 2020) or the utilization of pre-trained models (Junczys-Dowmunt et al., 2018). This category of approaches takes input text containing grammatical errors and employs an encoder-decoder architecture to produce corrected outputs.

Another methodology is based on sequence tagging, where the model generates an action for each input token to correct grammatical errors (Omelianchuk et al., 2020). Given the inherent nature of GEC, suggestions can be represented as a series of independent actions (keep, add, delete, replace), and grammatically correct text is derived by executing these actions in sequence. The following chapters reveal these two prominent families of methods.

2.2 Sequence-to-sequence error correction approaches

Bryant et al., 2023 conducted a comprehensive survey of state-of-the-art sequence-to-sequence (seq-to-seq) methodologies applied in the domain of GEC. Their findings indicate that the current advanced methods in GEC are similar to low-resource NMT systems configured in conditions with limited parallel data (Malmi et al., 2019). This insight implies that similar techniques also help improve neural GEC systems (Junczys-Dowmunt et al., 2018). The authors indicate a set of techniques that can help to perform better than traditional statistical machine translation systems and become a standard practice, even outperforming the quality of state-of-the-art systems. Key strategies to achieve state-of-the-art GEC tasks include:

- pre-training on large amounts of data, including synthetic, which is especially helpful in low-resource scenarios;
- increase model size;
- using multiple models together in an ensemble.

The advances of Transformer-based language models enable more adequate capture of syntactic phenomena Wei et al., 2021, making them capable GEC systems when little or no data is available. These systems can, however, become even more capable when exposed to a small amount of parallel data Mita and Yanaka, 2021. However, using synthetic data generated from error type tags led to significant performance gains on standard test sets and is effective in adapting systems for native English Stahlberg and Kumar, 2021.

More recently, with the advances in large pre-trained language models, direct model fine-tuning with GEC parallel data showed state-of-the-art performance Rothe et al., 2021. For example, in Rothe et al., 2021, authors adopted the mT5 (Xue et al., 2021) pre-trained on a corpus covering 101 languages as the base model for the multilingual GEC task. Another study Loem et al., 2023 shows no need for LLM fine-tuning to perform competitively in the GEC tasks, outperforming the Transformer model in all test sets. Zhang et al., 2023 proposed multi-task instruction fine-tuning that significantly improves LLMs ability on writing tasks, including GEC.

In Bryant et al., 2023, authors also note that while NMT is better at fixing complex errors, considering context, it has limitations, especially in needing a large amount of high-quality data (Rothe et al., 2021). Additionally, it is hard to understand the reason for the edit made by the model. However, NMT systems are advantageous because they perform grammatical error correction end-to-end.

Large Language Models present promising results in this context due to their proven effectiveness in various end-to-end NLP tasks, including GEC (Loem et al., 2023). However, the integration of LLMs into GEC is still an area in need of further exploration, particularly regarding the assessment of the capabilities of newer LLMs (Zhang et al., 2023). This exploration provides rationales for adapting models to specific domains. Notably, fine-tuning LLMs on domain-specific data incurs significant training expenses for domain adaptation and is heavily dependent on the quality of data (Zhang et al., 2023).

2.3 Sequence-to-tag approaches

In the realm of Natural Language Processing, for many tasks such as GEC, the input and output sentences may overlap significantly. Recent advancements have witnessed the emergence of innovative architectures, such as the Copy-Augmented Transformer, to address these challenges effectively (Hotate, Kaneko, and Komachi, 2020; Wan, Wan, and Wang, 2020). These models typically employ a full sequence approach, wherein the majority of tokens are directly transferred from input to output. While this method ensures lexical consistency, it often results in sub-optimal decoding speeds and increased computational loads due to the extensive vocabulary size. Moreover, this approach lacks an explanatory mechanism for its proposed target sequences.

To address these limitations, Awasthi et al., 2019 introduced a Parallel Iterative Edit model reducing decoding time for local sequence transduction tasks. In contradistinction to full-sequence models (Junczys-Dowmunt et al., 2018), this approach generates a sequence of edits, each aligned with input sentence tokens, focusing on specific edits rather than rewriting entire sentences.

Further refinement in this domain is evident in the sequence-to-edit and sequence tagging approaches, which accommodate multi-token edits (Omelianchuk et al., 2020; Tarnavskyi, Chernodub, and Omelianchuk, 2022). A primary advantage of these methods is their capacity for iterative refinement, enhancing the inference capabilities of parallel models.

Source	A	ten	years	old	boy	go	school
Target	A	ten-	year-	old	boy	goes to	school.
Tags	KEEP	KEEP, MERGE	NOUN, MERGE	KEEP	KEEP	VERB, APPEND	KEEP, APPEND

TABLE 2.1: Example task formulation of edit generation in the sequence-to-tag approach from Omelianchuk et al., 2020. Each tag represents the edit type and a replacement string.

For illustrative purposes, consider the sequence tagging approach applied to the sentence "A ten years old boy go school" and its corrected version "A ten-year-old boy goes to school." (Table 2.1). The necessary corrections can be represented as: [A → A], [ten → ten -], [years → year -], [old → old], [boy → boy], [go → goes to], [school → school.]. The optimal number of edits at the token level can be achieved by minimizing Levenshtein distance during tranduction, with each edit type categorized into labels such as KEEP, APPEND, DELETE, REPLACE, MERGE. Complex errors may require multiple actions; for instance, [years → year -] necessitates a REPLACE of the original token and a MERGE with dash . In Omelianchuk et al., 2020, authors recommend an iterative correction method, executing one edit per iteration for efficiency.

Despite its effectiveness, the sequence-tagging approach is limited by its token-level focus and the need for iterative execution to correct errors. Concurrently, alternative solutions like in Stahlberg and Kumar, 2020 proposal of edit-span operations have been developed. These operations involve a 3-tuple representing the action of replacing a span from positions $n-1$ to n in the source sentence with a replacement token and an explainable tag. Span-level edits offer a more compact representation than token-level edits and are simpler to learn due to the ease of capturing local dependencies within a span.

Certain approaches' methodologies exhibit a notable divergence from the architectures of LLMs. A key challenge lies in integrating these distinct methods into the framework of LLMs. In this context, the study by Kaneko and Okazaki, 2023 is particularly noteworthy. This research shows the feasibility of adapting LLMs to the sequence-to-edit task, a process markedly different from the more conventional sequence-to-tag methods. The primary distinction of this method from the sequence-to-tag approach is its ability to predict a set of edit spans, representing the changes in the target text relative to the source tokens. By omitting unedited tokens, which constitute the majority of the target text, the method significantly reduces the length of the target text and the inference time for local sequence transduction tasks.

2.4 Models ensembling

Ensembling, a prevalent methodology in machine learning, combines the outputs of multiple independently trained models. This technique is also significant in the domain of GEC, where diverse approaches exhibit distinct strengths and weaknesses. In Susanto, Phandi, and Ng, 2014, authors have demonstrated that the varying strengths of different GEC models can be used to enhance grammatical error correction through a method known as system combination. Recent advancements in GEC have largely been attributed to the ensembling of outputs from single models, as highlighted in the studies Awasthi et al., 2019; Omelianchuk et al., 2020; Tarnavskyi, Chernodub, and Omelianchuk, 2022.

In the context of GEC, ensembling typically entails averaging the probabilities or employing majority voting from individually trained GEC models. This process is applied when predicting the next token in a sequence-to-sequence approach or determining the edit tag in an edit-based approach.

The models selected for ensemble configurations generally possess similar properties, with minor variations such as differences in the random seed as noted by Stahlberg and Kumar, 2020, the choice of pre-trained model (Omelianchuk et al., 2020). Tarnavskyi, Chernodub, and Omelianchuk, 2022 have observed that the quality of corrections improves with the ensembling of output tag probabilities, indicating that a larger combination of models tends to yield better results. The application of majority vote aggregation for span-level edits has been shown to facilitate the combination of various models, achieving the best results at the time of publication on the BEA-2019 test benchmark.

Qorib, Na, and Ng, 2022 presented a novel paradigm ESC by reframing the combination of GEC systems as a binary classification task. This methodology is primarily based on the classification of isolated edits, without considering context or other edits into account, and can substantially enhance the quality of the ensemble GEC models. In further extension of the research Qorib and Ng, 2023 introduced an innovative approach for GEC quality estimation - GRECO. Their methodology diverges from the traditional practice of selecting the optimal correction from various GEC model outputs of different architectures (seq-to-seq or seq-to-tag). Instead, they proposed using a BERT-like pre-trained language model to evaluate the quality of edits on each iteration of the correction. The model, trained on the W&I train set with hyper-parameters tuning on BEA-2019 development set and the CoNLL-2013 test sets, evaluates the pairs of sentences – the original and the hypothesized correction – to verify whether an edit contributes to an improved grammatical structure. This technique was adapted specially for both CoNLL-2014 and BEA-2019 benchmarks and was marked a significant advancement in the field, achieving the latest state-of-the-art result on the CoNLL-2014 benchmark with an $\mathbf{F_{0.5}}$ score of 71.12%.

2.5 Results analysis

System	Type	CoNLL-2014 test	BEA-2019 test	BEA-2019 dev
Single model				
Marian Transformers EncDec, Junczys-Dowmunt et al., 2018	Seq-to-seq	56.25	-	-
PIE, BERT Awasthi et al., 2019	Seq-to-tag	59.7	-	-
Seq2Edits, Transformer-big Stahlberg and Kumar, 2020	Seq-to-tag	58.6	-	48
BERT-fuse GEC Kaneko et al., 2020	Seq-to-seq	62.6	65.6	-
GECToR Omelianchuk et al., 2020	Seq-to-tag	65.3	72.4	-
Seq-tag, Tranformer-big Stahlberg and Kumar, 2021	Seq-to-seq	66.6	70.4	-
gT5, T5-xxl Rothe et al., 2021	Seq-to-seq	**68.75**	**75.88**	-
Large Sequence Tagger, RoBERTa, Tarnavskyi, Chernodub, and Omelianchuk, 2022	Seq-to-tag	-	73.21	55.8
EditSpans LLM (LLaMA) Kaneko and Okazaki, 2023	Decoder-only	68.2	-	-
LLaMA-7B-GEC Zhang et al., 2023	Decoder-only	65.2	-	54.6
LLaMA-13B-GEC Zhang et al., 2023	Decoder-only	67	-	56.1
Zero-shot LLM				
GPT-3 Loem et al., 2023	Decoder-only	**57.06**	**57.41**	-
Zero-shot ChatGPT Fang et al., 2023	Decoder-only	50.3	34.4	-
Zero-shot CoT ChatGPT Fang et al., 2023	Decoder-only	51.7	36.1	-
Ensemble methods				
PIE, BERT Awasthi et al., 2019	Seq-to-tag	61.2	-	-
Seq2Edits Stahlberg and Kumar, 2020	Seq-to-tag	62.7	70.5	-
Ensemble of BERT-fuse GEC Kaneko et al., 2020	Seq-to-tag	65.2	69.8	-
GECToR Omelianchuk et al., 2020	Seq-to-tag	66.5	73.7	-
Seq-tag, Tranformer-big Stahlberg and Kumar, 2021	Seq-to-tag	**68.3**	74.9	-
Large Sequence Tagger, Tarnavskyi, Chernodub, and Omelianchuk, 2022	Seq-to-tag	-	**76.05**	-
Edit scorers				
ECS Qorib, Na, and Ng, 2022	Scorer	69.51	79.9	63.09
GRECO Qorib and Ng, 2023	Scorer	71.12	80.84	63.4
Chat-LLaMA-2-13B-FT (Ours)	Decoder-only	67.87	73.11	56.43

"-" - denotes to no data provided in the original paper.

TABLE 2.2: Comparison of the $\mathbf{F_{0.5}}$ scores for GEC systems on CoNLL-2014, BEA-2019 benchmarks.

In the current discourse, we examine the recent advancements in state-of-the-art systems for Grammatical Error Correction, particularly those developed within the last few years, and delineate the innovative methodologies that have enhanced their performance beyond preceding efforts. Table 2.2 consolidates the assessments conducted on models using GEC benchmarks: BEA-2019 (Bryant et al., 2019) and CoNLL-2014-test (Ng et al., 2014).

A significant observation from Table 2.2 is the prevalence of evolved sequence-to-tag methods, notably PIE (Awasthi et al., 2019) and GECToR (Omelianchuk et al., 2020). Omelianchuk et al., 2020 advanced this field by integrating a pre-trained language model, such as BERT, into a sequence tagging framework. Enhancements such as increasing the size of the pre-trained language model and introducing additional mechanisms for selecting final edits – either through edit-scoring or majority voting – have been shown to augment baseline performance (Tarnavskyi, Chernodub, and Omelianchuk, 2022).

Recent developments in Large Language Models have proven to be significant. Zhang et al., 2023 have shown that fine-tuning LLMs with extensive in-domain data significantly outperforms multi-task fine-tuning strategies, achieving results that are on par with established baseline models on both the CoNLL-2014 and BEA-2019 benchmarks. On the other hand, research by Loem et al., 2023 and Fang et al., 2023 suggests that while zero-shot and few-shot approaches show strong error detection capabilities with fluently corrected text, they underperform across most error types and fall short of delivering high-quality outcomes in Grammatical Error Correction.

In addressing a different aspect of GEC, the GRECO approach by Qorib and Ng, 2023 for GEC quality estimation not only provides more accurate estimates but achieves state-of-the-art results, further contributing to the advancements in the field.

Notably, most approaches have predominantly utilized small—to medium-sized language models for specific tasks such as edit generation or scoring. However, the application of Large Language Models for both edit generation and edit scoring in ensembles in this domain remains underexplored. Our research will focus on applying LLMs for grammatical error correction, recognizing the potential for significant advancements in this area.

Chapter 3

Data review and evaluation

Data and the evaluation of model performance metrics serve as foundational components for the majority of NLP tasks, including GEC. The literature review section acknowledges that GEC can be considered as low-resource machine translation task (Junczys-Dowmunt et al., 2018). Researchers showed that increasing dataset size is critical for achieving state-of-the-art results. The acquisition of high-quality, annotated data presents a significant challenge. Therefore, we observe GEC datasets in English, including both human-annotated and synthetic, that can be used in this research.

Moreover, this section describes most widely recognized evaluation metrics in GEC - the MaxMatch (M2) scorer (Dahlmeier and Ng, 2012) and ERRANT (Bryant, Felice, and Briscoe, 2017). It addresses the issue of metric reliability, especially concerning their correlation with human judgments, and describes the challenges associated with deriving definitive conclusions.

3.1 Datasets

Dataset	Part	# Sent.	# Toks.	Error types	Domain
Lang-8	Train	1.03m	11.8m	28	Essays
NUCLE	Train	57.1k	1.16m	28	Essays
CoNLL-2013	Dev/Test	1.4k	28.2k	28	Essays
CoNLL-2014	Test	1.3k	29.2k	28	Essays
FCE	Train	28.3k	454k	71	Exams
	Dev	2.2k	34.7k	71	Exams
	Test	2.7k	41.9k	71	Exams
cLang-8	Train	2.37M	28.0M	58	Essays
Troy-1BW	Train	1.2M	30.88M	-	General
Troy-Blogs	Train	1.2M	21.49M	-	General

TABLE 3.1: Statistics of GEC datasets used in this work for training and evaluation.

Lang-8. Corpus of Learner English, initially introduced by Mizumoto et al., 2012; Tajiri, Komachi, and Matsumoto, 2012, represents a subset of the broader multilingual Lang-8 Learner Corpus. This corpus aggregates texts from different domains and proficiency levels. Despite the corpus being officially designated as the official training dataset for BEA-2019 (Bryant et al., 2019) shared task and its status as one of the largest corpora accessible to the public, the Lang-8 Corpus was annotated by fellow users, as opposed to professional annotators, which impacts the consistency and reliability of the data.

NUCLE. The National University of Singapore Corpus of Learner English (Dahlmeier, Ng, and Wu, 2013), officially used as a training corpus in CoNLL-2013 and CoNLL-2014 Ng et al., 2013; Ng et al., 2014 as well as BEA-2019 Bryant et al., 2019 shared tasks. It has been created from the essays authored by undergraduate students at the National University of Singapore (NUS) who required support in English as a second language (L2). Predominantly aligned with the C1 proficiency level of the Common European Framework of Reference for Languages (CEFR), these essays cover a broad spectrum of subjects such as technology, healthcare, and finance. Each essay underwent correction by a singular annotator, who identified and categorized every modification under a comprehensive framework comprising 28 distinct error types.

The test subset for the CoNLL-2013 contains essays on topics of surveillance technology and population aging and CoNLL-2014 - genetic testing and social media. Notably, the CoNLL-2014 test set (marked in research as **CoNLL-2014-test**) contains 18 annotated references collected by two independent annotators and Bryant and Ng, 2015; Sakaguchi et al., 2016.

While the CoNLL-2013 dataset occasionally serves as a development set, the CoNLL-2014 dataset remains among the most frequently used benchmark test sets for GEC tasks. However, a notable limitation of the CoNLL-2014 test set is its lack of diversity; it comprises essays exclusively penned by a relatively homogeneous group of learners, focusing only on two distinct topics. This specificity could potentially restrict the generalizability of findings derived from its use as a benchmark.

FCE. The First Certificate in English corpus Yannakoudakis, Briscoe, and Medlock, 2011 is a publicly available subset of the Cambridge Learner Corpus (CLC) Nicholls, 1999 containing writings from international learners of English as a second language (L2 learners). Each text submitted as a short essay, letter, or description is scored and corrected by a singular annotator. Each edit in correction is classified within 88 distinct error types, although only 71 unique error types are represented within the FCE subset, which makes this corpus extremely useful for error detection tasks Yuan et al., 2021. The FCE corpus was used as the training dataset of BEA-2019 and Helping Our Own (HOO) 2012 Dale, Anisimoff, and Narroway, 2012 shared tasks.

The Write & Improve (**W&I**) and **LOCNESS** corpus Bryant et al., 2019, also known as BEA-2019, contains essays authored by international learners across all proficiency levels (A1-C2) and native British and American English undergraduate students. This corpus was designated as the official dataset for training, development (market in research as **BEA-2019-dev**), and testing within the context of the BEA-2019 shared task Bryant et al., 2019. Its construction aimed to achieve a more equitable distribution across proficiency levels—beginner, intermediate, advanced, and native—than observed in other corpora, ensuring a roughly equal representation of sentences from each category. Essays within the training and development sets were subject to correction by a single annotator. Conversely, essays within the test set were reviewed by five annotators, yielding five sets of parallel reference annotations. Although edits were explicitly defined, they were not manually classified; instead, error types were automatically assigned utilizing the ERRANT framework Bryant, Felice, and Briscoe, 2017. The test set references are not publicly accessible to ensure equality of assessment of GEC systems, facilitating fair and consistent comparisons across research efforts.

cLang-8 (Rothe et al., 2021) corpus is a large cleaned version of the Lang-8 Mizumoto et al., 2012 corpus. The original Lang-8 corpus contains user-annotated corrections that frequently contain unnecessary paraphrasing and erroneous or incomplete corrections. Model-based scoring was used to select the best pairs of source and corrected sentences.

Troy-1BW and **Troy-Blogs** (Tarnavskyi, Chernodub, and Omelianchuk, 2022), synthetic datasets produced from the One Billion Word Benchmark Chelba et al., 2014 and Blog Authorship Corpus (Schler et al., 2006) with using model ensembles.

JFLEG - The Johns Hopkins Fluency-Extended GUG corpus (Napoles, Sakaguchi, and Tetreault, 2017) consists of 1,500 sentences randomly sampled from essays by L2 learners. These sentences have been revised for fluency beyond the minimal necessary grammatical corrections by crowdsourced annotators on Amazon Mechanical Turk, with each sentence having four reference versions. We excluded this dataset from our research due to its relatively small size, the inclusion of non-grammatical corrections, and the fact that the corrections were not made by professionals.

3.2 Evaluation

Evaluating the performance of models is a crucial aspect of any machine learning task, including Grammatical Error Correction. There are two main approaches to evaluating GEC systems: reference-based and reference-less. This section outlines the most commonly used reference-based evaluation metrics in the field. These include the MaxMatch (M2) scorer (Dahlmeier and Ng, 2012) and ERRANT (Bryant, Felice, and Briscoe, 2017), which are frequently cited in GEC research.

3.2.1 MaxMatch (M2) Scorer

Source	A	ten	years		old	boy	go		school	
Hypothesis	A	ten	years		old	boy	go		market	
Reference	A	ten	years	-	old	boy	goes	to	school	.
Edit label	TP	TP	TP	FN	TP	TP	TP	FN	FP	FN

TABLE 3.2: Example error labels for M2 metrics calculation.

The MaxMatch (M2) scorer, a primary evaluation tool in GEC research, uses $F\beta$-score for system comparison based on hypothesis edits and human-annotated reference edits. This reference-based metric assesses (Table 3.2) True Positives (TPs), False Positives (FPs), and False Negatives (FNs) to calculate Precision (P), Recall (R). Research conducted by Grundkiewicz, Junczys-Dowmunt, and Gillian, 2015 and Chollampatt and Ng, 2018 found that the $\mathbf{F_{0.5}}$ score, that weights precision twice as much as recall, exhibits a higher correlation with human judgment compared to other GEC metrics, establishing its relevance and reliability in assessing grammatical correctness on the public shared tasks. A unique feature of the M2 scorer is its use of Levenshtein alignment to dynamically explore various combinations of edits, thereby maximizing the match between hypothesis and reference edits, which addresses limitations seen in prior metrics.

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F_\beta = (1 + \beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R}$$

The method has several limitations. The first issue is that overlapping edits can be unclear, which might cause even correct edits in the hypothesis to be counted as errors if they aren't presented in the same way in the references. To address this, using multiple references can help find the best match. Another limitation is that the method can't evaluate systems based on the type of errors they can fix.

3.2.2 ERRANT

The alternative to M2 scorer is ERRANT, which measures performance via an edit-based F-score but extends its capabilities by categorizing edits into error types and is considered as the primary metric in the BEA-2019 (Bryant et al., 2019) benchmark. It employs a Damerau-Levenshtein alignment algorithm from Bard, 2006 to extract and classify edits according to a rule-based framework. The framework can operate on different levels like edit operations types (missing, unnecessary or replacement), or error types. The advantage of the method is its ability to evaluate GEC systems based on error types, offering insights into system performance on specific grammatical issues. Although initially developed for English, ERRANT has been adapted for other languages, showcasing its flexibility and applicability in a broader linguistic context.

3.2.3 Discussion on Metric Reliability

Metric	r	ρ
ERRANT $\mathbf{F_{0.5}}$	0.919	0.887
M2 $\mathbf{F_{0.5}}$	0.860	0.849
GLEU	0.838	0.813
I-measure	0.819	0.839

TABLE 3.3: Pearson r and Spearman ρ correlation coefficients for GEC metrics on FCE evaluation set, presented in Napoles, Nădejde, and Tetreault, 2019 Table 8.

Beyond M2 and ERRANT, several other metrics have been developed that rely on references or do not require them at all (Bryant et al., 2023), like *I-measure* or *GLUE*. Each metric has its own advantages and disadvantages. Finding the "best" metric that matches human judgments is still an open question (Napoles, Nădejde, and Tetreault, 2019). However, Table 3.3 shows the correlation between GEC metrics and human evaluations, helping us find the metric that best reflects human judgment. The ERRANT $\mathbf{F_{0.5}}$ and M2 $\mathbf{F_{0.5}}$ metrics have the highest correlation and are considered the primary metrics in this research. It's important to note that while human judgments are common and best benchmark GEC systems evaluating, they are also subjective and should be approached with well-defined guidelines (Bryant et al., 2023).

Considering this, there is no empirical evidence to favor one metric over another. However, in practice, the most widely used benchmarks in grammatical error correction tasks are:

- CoNLL-2014 is evaluated with the M2 scorer;
- BEA-2019 is evaluated with ERRANT.

We primarily use both benchmarks due to historical practices, which helps ensure consistency when comparing our systems with the current state-of-the-art.

Chapter 4

Experiments with Large Language Models

Recent studies demonstrate that Large Language Models, such as ChatGPT, LLaMA (Touvron et al., 2023), exhibit remarkable proficiency in grammar correction (Fang et al., 2023; Zhang et al., 2023). A notable example is the study by Loem et al., 2023, which explored the capabilities of the proprietary LLM GPT-3 in the GEC task without fine-tuning. Another study by Rothe et al., 2021 highlighted the effectiveness of fine-tuning LLMs, specifically the mT5 model with up to 11 billion parameters, in establishing new baselines for GEC and defines yet simple configuration of GEC systems.

We investigated the capabilities of LLMs in GEC through various settings, identifying three main approaches (Brown et al., 2020):

- Supervised Fine-Tuning (SFT) updates the weights of a pre-trained model through training on a task-specific supervised dataset, typically ranging from thousands to hundreds of thousands of labeled examples. The primary advantage is improved performance across many benchmarks, while the main disadvantages include the need for extensive datasets, potential overfitting, strong dependency on data quality, and the high costs associated with fine-tuning.

- Zero-Shot (ZS) refers to the method in which the model is prompted to solve a task at inference time without weight adjustments. This approach maximizes convenience and potential robustness, minimizing biases present in training datasets. However, it requires accurate task formulation, which can be challenging without prior examples. In GEC, a precise definition is essential, as errors can be corrected through direct fixes or paraphrasing. Despite its limitations, zero-shot resembles how humans approach unfamiliar tasks, such as translation, based only on instructions.

- Few-Shot (FS) resembles zero-shot but introduces several examples during the task setup. This method enhances output quality through contextual conditioning, even with vague task definitions. However, it still underperforms compared to state-of-the-art fine-tuned models and requires task-specific data.

This study concentrates on evaluating the zero-shot and supervised fine-tuning approaches in two modes, full and parameter-efficient tuning settings, to assess their effectiveness and find the trade-off between model quality and the time required for training. We exclude the few-shot setting due to its heavy reliance on data quality, and it will be considered in separate studies.

We chose three open-source Large Language Models that have been trained across tasks involving language understanding, reasoning, and safety. These include Mistral v0.2 7B (Instruct-Mistral-7B(13B)-ZS, Jiang et al., 2023), Gemma of size 2B and

7B (Instruct-Gemma-7B(2B)-ZS, Team et al., 2024), and LLaMA2 of 7B and 13B sizes (Chat-LLaMA-2-7B(13B)-ZS, Touvron et al., 2023). These models were selected because they demonstrate comparable performance within similar sizes (Team et al., 2024) on question-answering and reasoning tasks.

4.1 Zero-shot prompting

Recent studies Coyne and Sakaguchi, 2023; Fang et al., 2023; Loem et al., 2023 have explored the effectiveness of prompt-based methods in GEC benchmarks. These studies indicate that hard prompt tuning significantly enhances GPT-3's performance in both zero-shot and few-shot settings, with consistency improving as the model is exposed to more examples. Tailoring prompts to include specifics like the language proficiency levels of L2 learners and the types of errors to be corrected can mitigate the common issue of LLMs over-correcting, thereby increasing recall but reducing precision (Fang et al., 2023). To our knowledge, existing research has investigated the capabilities of proprietary models only in the zero-shot setting, and there has been no exploration of open-source Large Language Models like LLaMA (Touvron et al., 2023), Gemma (Team et al., 2024), or Mistral (Jiang et al., 2023). This section aims to assess the capabilities of these instruction-tuned models on the GEC task using various prompts.

In the zero-shot setting, we evaluated the ability of open-source LLMs to perform GEC tasks without prior examples. We utilized model-specific templates and task instruction prompts as follows:

```
You are helpful AI assistant.
# Task
{instruction}

# Output format
Answer with corrected text only. If there are no errors,
respond with the original text.

# Prediction
Text: {text}
Corrected text:
[____]
```

We use the task preamble (system prompt) to condition model behavior as a writing assistant and enable the generation of output in a convenient for post-processing format.

We vary the task formulation to find its influence on final metrics (Table 4.1).

We use the generation parameters during inference to $temperature = 10^{-3}$ and $beamsize = 1$ to eliminate stochastic generation and ensure the reproducibility of our experiments. We performed response parsing to extract corrected text from the model's output or retain the input text if the model classifies it as error-free and requiring no correction.

Table 4.2 presents an ablation study evaluating open-source LLMs on the selected GEC benchmarks: CoNLL-2014-test and BEA-2019-dev.

Model performance depends on how tasks are formulated, which aligns with findings from Loem et al., 2023 that LLMs often exhibit over-correction, leading to high recall but lower precision, as seen with prompt #2. Directing the model to

Prompt ID	**Prompt text**
#1	*Fix grammatical errors for the following text . Keep only one variant .*
#2	*Rewrite this text to make it grammatically correct .*
#3	*Rewrite the text to fix any grammatical errors .*
#4	*Correct the grammar mistakes in the following text .*
#5	*Rewrite the text . The output text should not contain any grammatical or spelling mistakes .*
#6	*Fix all grammatical errors , do not rephrase .*
#7	*Fix only grammatical errors precisely.*
#8	*Precisely fix grammatical errors :*
#9	*Revise the following sentence with proper grammar*
#10	*Correct grammatical errors in this sentence*
#11	*Revise grammatical mistakes in the following text.*
#12	*Revise mistakes in the following text written by a beginner learner with a lot of mistakes .*
#13	*Revise mistakes in the following text written by a advanced learner with a few of mistakes .*

TABLE 4.1: GEC Task formulation for Zero-Shot setting.

	CoNLL-2014-test			**BEA-2019-dev**		
Prompt ID	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
1	50.85	50.03	50.68	32.57	41.08	33.98
2	42.17	**54.76**	44.20	22.40	40.99	24.63
3	46.81	52.03	47.77	28.33	41.56	30.26
4	50.94	50.54	50.86	32.38	41.36	33.85
5	44.39	51.76	45.69	25.68	40.34	27.70
6	**53.62**	48.39	**52.61**	**35.64**	40.22	**36.47**
7	51.4	50.27	51.17	33.01	40.88	34.33
8	50.03	50.20	50.00	31.78	41.62	33.36
9	45.16	52.52	46.46	26.48	**41.90**	28.58
10	51.62	49.77	51.24	32.94	40.89	34.27
11	51.17	50.80	51.10	31.82	41.43	33.37
12	50.44	51.14	50.58	30.76	40.50	32.32
13	51.19	50.71	51.10	32.06	40.28	33.34

TABLE 4.2: Comparison of the zero-shot setting for Chat-LLaMA2-7B-ZS on CoNLL-2014-test and BEA-2019-dev.

avoid rephrasing the input sentence (prompt #6) or to focus exclusively on correcting grammatical errors (comparison of prompt #6 and prompt #7) has improved precision by 2 percentage points across both benchmarks.

Moreover, adding specific details to the task formulation, such as the learner's proficiency level (prompt #12 and #13) or the extent of required edits, enhances the precision (prompt #7 and #8) of the outputs (Table 4.2 and A.2). This approach was used for GPT-3 models in Loem et al., 2023 and is applicable for open-source foundational LLMs like Chat-LLaMA-2-7B-ZS.

To substantiate these observations, we use prompt #6 with selected open-source LLMs, including models of different sizes like Chat-LLaMA-2-13B-ZS and other architectures such as Instruct-Mistral-7B-ZS and Instruct-Gemma-2B-ZS, to evaluate their performance under comparable conditions.

		CoNLL-2014-test			**BEA-2019-dev**		
System	**Prompt #**	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
Chat-LLaMA-2-7B-ZS	6	**53.62**	48.39	52.61	**35.64**	40.22	**36.47**
Chat-LLaMA-2-13B-ZS	6	51.12	**54.47**	**52.75**	32.47	**44.14**	34.28
Instruct-Mistral-7B-ZS	6	40.44	**56.02**	42.83	21.29	**31.66**	22.78
Instruct-Mistral-7B-ZS	7	**43.21**	53.33	**44.9**	21.94	31.89	**23.4**
Instruct-Gemma-2B-ZS	6	35.39	45.55	37.04	15.35	29.65	16.99
Instruct-Gemma-7B-ZS	6	**50.3**	**53.32**	**50.87**	**31.44**	**42.92**	**33.22**

TABLE 4.3: Comparison of the zero-shot setting for LLaMA2, Mistral and Gemma LLMs on CoNLL-2014-test, BEA-2019-dev.

The comparative performance of various open-source LLMs, as shown in Table 4.3, illustrates distinct outcomes based on model size and architecture. For instance, while the larger Chat-LLaMA-2-13B-ZS model shows an improvement in recall, it maintains comparable precision with its 7B version, indicating that increased model size can enhance recall without a significant drop in precision.

The Instruct-Mistral models, particularly the 7B version, exhibit superior recall on CoNLL-2014-test benchmark, while BEA-2019 became a challenging task, aligning with their design goal to support complex writing tasks (Jiang et al., 2023) and content generation. At the same time, the smaller Gemma 2B model demonstrates limitations in generating high-quality grammatical corrections, likely due to its smaller size and reduced capacity relative to larger models.

In conclusion, our findings prove the critical role of precise task formulation and model-specific strategies in optimizing LLM performance for grammatical error correction. This led to a change in up to 11.8% of $\mathbf{F_{0.5}}$ on CoNLL-2014-test and BEA-2019-dev benchmarks (Table A.2). Specific prompts have proven effective across various models, suggesting a robust approach to improving LLM accuracy in real-world applications.

4.2 LLM supervised fine-tuning

4.2.1 Full weights fine-tuning

LLM finetuning shows exceptional performance on the GEC task (Rothe et al., 2021) and is a required stage to achieve SOTA results (Zhang et al., 2023). In this section, we explored open-source LLM fine-tuning settings to find the best configuration that minimizes the number of fine-tuning stages and defines the training data, dataset size, instruction tuning, and other parameters that may influence the model quality.

We fine-tune the official version of LLaMA2 with the Huggingface Transformers[1] toolkit to conduct 1000–1200 updates with 250 warm-up steps, a batch size of 8, and a learning rate of 10^{-5}. During training, we optimize LLM to output the reference response via cross-entropy loss on the next token prediction task. Considering the time and computational resources, we perform parameter-efficient fine-tuning and full-model fine-tuning to show the trade-off between the training costs and the result in model quality. Specifically, we mainly utilize the low-rank adaptation (LoRA) Hu et al., 2021 technique for computational effectiveness in some of our experiments. We also compare LoRA with full model fine-tuning. All experiments are carried out on 4 Nvidia A10G 24GB GPUs.

Dataset combination

In the following experiments, we fine-tune the LLaMA2 model on three historically most popular datasets we defined in the previous chapter: NUCLE, W&I, and cLang-8. We use several setups to identify the best dataset in single dataset training and also combine datasets in different proportions.

The training results on the joined dataset can be seen in Table 4.4.

Results from the ablation study of the best dataset combination search for fine-tuning LLM show the significant change in $F_{0.5}$ scores for a single dataset and the combination. In single datasets experiments, the model trained on W&I outperforms the one trained on NUCLE on 6 $F_{0.5}$ points, making this dataset highly valuable in our experiments. Using only true positives, samples that contain corrections

[1] https://github.com/huggingface/transformers

Model	Datasets			CoNLL-2014-test			BEA-2019-dev		
	NUCLE	**W&I**	**cLang-8**	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
LLaMA-2-7B-FT	-	34.3k	-	68.66	**54.27**	65.20	57.90	**48.63**	55.77
LLaMA-2-7B-FT	-	-	2.3M	67.25	50.44	63.05	57.99	42.11	53.93
LLaMA-2-7B-FT	57.1k	34.3k	-	72.45	46.98	65.37	58.00	45.82	55.07
Chat-LLaMA-2-7B-FT	57.1k	-	-	70.39	36.31	59.42	50.72	24.51	41.79
Chat-LLaMA-2-7B-FT	-	34.3k	-	70.45	52.59	65.97	**59.19**	47.81	**56.50**
Chat-LLaMA-2-7B-FT	-	34.3k	100k	68.94	52.78	64.96	57.94	45.53	54.94
Chat-LLaMA-2-7B-FT	57.1k	34.3k	48k	**75.40**	46.84	**67.20**	58.26	46.03	55.32
Chat-LLaMA-2-7B-FT	8k, TP	8k, TP	24k, TP	68.01	52.84	64.32	53.94	46.03	52.15

TABLE 4.4: A search of the best training dataset combination for fine-tuning Large Language Models. For fine-tuned models, different training dataset combinations were evaluated: Here, "TP" ("true positives") denotes when only the dataset's samples containing corrections are used.

significantly improve recall but force the model to overcorrect, reflecting a drop in precision. The best model in all sets of experiments with data combination was fine-tuned on all used datasets. We vary the combination of several datasets and use only a random subsample of the cLang-8 dataset in our experiments to ensure the model also uses all data from other datasets. Still, we don't find any single-model system approach dominant across all benchmarks. In our next experiments, we take only the W&I dataset as training to avoid extra complexity with mixing the dataset, as it does not significantly improve quality across both benchmarks.

Model size

Next, we investigate the influence of using instruction and no-instruction tuning setups for full weights fine-tuning. Not surprisingly, our results (Table 4.5) indicate that instructions work better for "Chat" versions of models adapted to chatbot use-case.

At the same time, we varied the model size, and the experiments showed that the bigger model performed better on both benchmarks.

Model	Instructions	CoNLL-2014-test			BEA-2019-dev		
	are used	**Precision**	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
LLaMA-2-7B-FT	No	**69.33**	50.26	64.44	**59.45**	46.29	**56.25**
LLaMA-2-7B-FT	Yes	68.66	**54.27**	**65.20**	57.9	**48.63**	55.77
Chat-LLaMA-2-7B-FT	No	67.53	**53.59**	64.19	58.00	47.37	55.51
Chat-LLaMA-2-7B-FT	Yes	**70.45**	52.59	**65.97**	**59.19**	**47.81**	**56.50**
LLaMA-2-7B-FT	Yes	68.66	54.27	65.20	57.9	48.63	55.77
LLaMA-2-13B-FT	Yes	**71.49**	**55.67**	**67.65**	**60.28**	**49.26**	**57.69**
Chat-LLaMA-2-7B-FT	Yes	70.45	52.59	65.97	**59.19**	47.81	56.50
Chat-LLaMA-2-13B-FT	Yes	**72.35**	**54.48**	**67.90**	59.04	**48.73**	**56.64**

TABLE 4.5: Ablation study on instructions' usage in fine-tuned on W&I dataset Large Language Models.

The model trained using a full weights fine-tuning approach demonstrates performance comparable to other state-of-the-art GEC systems, as shown in (Table 2.2), while utilizing significantly less training data—34k from the W&I train set compared to 800k for gT5 (Rothe et al., 2021). However, this approach is computationally expensive (Table A.6). In the following section, we will explore parameter-efficient tuning as an alternative method that reduces computation time by limiting the number of trainable parameters. This approach aims to provide a balance between model quality and the number of trained parameters.

4.2.2 Parameter efficient fine-tuning

Transformers have significantly enhanced tracking performance but are resource-intensive, particularly in Large Language Models. High-performance Transformer models often require expensive computational resources, including multiple top-tier data-center GPUs and time to perform training. To mitigate these demands, the Parameter-Efficient Fine-Tuning (PEFT) approach was developed for Large Language Models, which face prohibitive costs with full fine-tuning (Ding et al., 2022). PEFT techniques, such as fine-tuning a small subset of parameters while keeping the rest unchanged, drastically cut computational and storage expenses.

In this section, we explore the use of PEFT in training LLMs for the Grammatical Error Correction task. Among the various PEFT methodologies, we focus on Low-Rank Adaptation (LoRA) (Hu et al., 2021). LoRA enhances parameter efficiency by integrating trainable rank decomposition matrices into specific dense layers of the model, maintaining performance comparably with other PEFT strategies like adapters (Houlsby et al., 2019) and prompt tuning (Li and Liang, 2021) without increasing inference time. Below, we detail the parameter update process derived from this fine-tuning approach.

$$W_{finetuned} = W_{pretrained} + \Delta W = W_{pretrained} + A \cdot B,$$

where A and B are the rank decomposition matrices and their product approximates ΔW - the trainable parameters.

Under Transformer architecture, certain weight matrices are linked with the self-attention mechanism, namely query W_q, key W_k, value W_v, and outputs W_o weight matrices, besides two more in the Multi-Layer Perceptron (MLP) module.

Following the recommendation of the original LoRA paper (Hu et al., 2021), we conducted experiments by optimizing the W_q and W_v matrices with a rank of 8 and alpha of 16. We also enabled fine-tuning of all available parameters W_q, W_k, W_v, W_o for the LLaMA2, Gemini, and Mistral open-source LLMs. For model conditioning, we used a model-specific template and system prompt as defined in the Zero-shot prompting section, employing prompt #6 (see Table 4.1). We found that the absence of a system prompt led to a significant drop in generation quality for all models adapted for instruction settings. The training was performed using the W&I dataset, with settings including 2000 updates, 250 warm-up steps, a batch size of 8, and a learning rate of 10^{-5}, assuming models of the same size have similar training configurations.

Model	**Adapters are used**	**CoNLL-2014-test**			**BEA-2019-dev**		
		Precision	**Recall**	**$F_{0.5}$**	**Precision**	**Recall**	**$F_{0.5}$**
Chat-LLaMA-2-7B-FT	W_q, W_v	59.93	45.29	56.29	48.88	32.58	44.44
Chat-LLaMA-2-7B-FT	W_q, W_k, W_v, W_o	**63.42**	**50.83**	**60.42**	**53.49**	**44.2**	**51.33**
Chat-Gemma-7B-FT	W_q, W_k, W_v, W_o	**63.87**	52.59	61.24	51.68	44.34	50.02
Chat-Mistral-7B-FT	W_q, W_k, W_v, W_o	60.83	**53.59**	**61.48**	**53.58**	**46.27**	**51.94**
Chat-LLaMA-2-7B-FT	Full	70.45	52.59	65.97	59.19	47.81	56.50

TABLE 4.6: Comparing the impact of fine-tuning LoRA adapters on model quality.

The results show that increasing trainable parameters improve the model performance significantly (from 4 to 7 points) for Chat-LLaMA-2-7B-FT models. At the same time, full model fine-tuning significantly outperforms the LoRA settings by 5 percentage points on both CoNLL-2014-test and BEA-2019-dev benchmarks, suggesting that this optimization may not be worthwhile.

4.3 Direct preference optimization

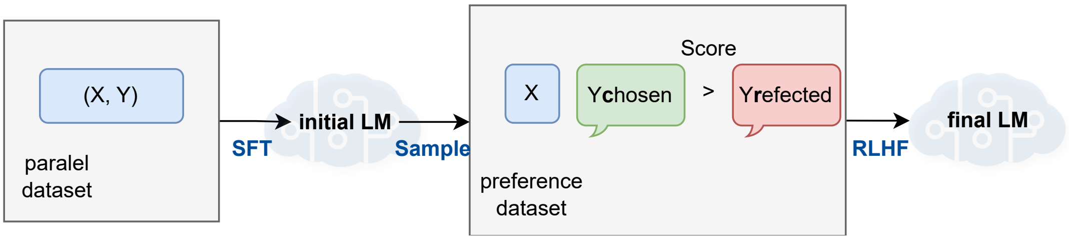


FIGURE 4.1: Reinforcement learning from Human Feedback on GEC.

In the realm of fast Large Language Model development, aligning models with human preferences and ethical standards is crucial for creating practical, controllable, and socially accepted writing assistants (Ouyang et al., 2022). Reinforcement Learning from Human Feedback (RLHF) is a method developed to enhance LLM performance by integrating human feedback (Touvron et al., 2023; Team et al., 2024). The RLHF process involves three main steps: (1) supervised fine-tuning, (2) reward modeling, and (3) RL fine-tuning, using datasets D_{SFT} for initial supervised training (1) and preference dataset D for refining the model π_θ based on human feedback.

RL fine-tuning requires a reward model r_Φ, which learns from the preference dataset D. The objective is to maximize expected rewards while minimizing deviations from a reference model π_{ref} (the model obtained on SFT stage) before RL fine-tuning, using policy gradient methods such as proximal policy optimization (Schulman et al., 2017). The optimization objective is defined as follows (Rafailov et al., 2023):

$$\max_\theta \mathbb{E}_{x\sim D}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta D_{KL}\left[\pi_\theta(\cdot|x)\|\pi_{\text{ref}}(\cdot|x))\right]\right] \tag{4.1}$$

where D_{KL} is Kullback–Leibler divergence of a distribution p from another distribution q, defined as $D_{KL}(p,q) = \mathbb{E}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right]$

Here, β is a hyper-parameter that controls the penalty for the deviations from the reference model π_{ref}.

The prior RLHF methods require a large diverse preference dataset and trained reward model to perform fine-tuning, making a problem computationally expensive and unstable in training (Rafailov et al., 2023). Direct Preference Optimization (DPO) joins the reward modeling and RL fine-tuning into a single phase, focusing on aligning the model π_θ directly with preference D data without needing a separate reward model (Rafailov et al., 2023). User preferences are directly incorporated into the optimization process, making it much easier to use and understand.

By eliminating the need for a separate reward model, DPO significantly reduces the computational cost of fine-tuning. With DPO, users have a more direct influence on the LLM's behavior. They can directly express their preferences, guiding the model towards specific goals and ensuring it aligns with their expectations. Due to its simpler structure and direct optimization approach, DPO often achieves desired results faster than RLHF (Rafailov et al., 2023).

The DPO objective function aims to maximize the ratio of probabilities for the chosen responses, optimizing the LM to imitate human preferences:

$$L_{DPO}(\theta) = \mathbb{E}_{(x,y_c,y_r)\sim D}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_c|x)}{\pi_{\text{ref}}(y_c|x)} - \beta\log\frac{\pi_\theta(y_r|x)}{\pi_{\text{ref}}(y_r|x)}\right)\right] \tag{4.2}$$

where β is a hyper-parameter and has a similar role as in Eq. 4.1 to control deviation of π_θ from reference π_{ref} model, y_c, y_r are **c**hosen and **r**ejected generated text for x input of preference dataset. DPO simplifies the optimization process by not requiring the generation of responses y from π_θ during training and scoring them with a reward model or incorporating human feedback, unlike the standard RL fine-tuning of Eq. 4.1.

4.3.1 Reformulating GEC as Preference Tuning task

Creating a preference dataset for RLHF involves selecting pairs of model outputs, where "chosen" outputs contain preferred corrections for the GEC task, and "rejected" outputs contain incorrect corrections. Gathering this dataset manually can be labor-intensive. To simplify this process, we propose using preference tuning, where human feedback is replaced by automatic metrics to evaluate and select pairs of chosen and rejected samples based on their grammatical error correction quality. In this work, we use the GRECO Quality Estimation model (Qorib and Ng, 2023) and the Scribendi reference-less metric (Islam and Magnani, 2021) as a scoring approach for model outputs. These metrics help identify the most accurate corrections and aim to achieve state-of-the-art results on GEC benchmarks.

We hypothesized that the language model, fine-tuned in the previous stage, would perform the best error correction using a greedy generation strategy. However, the experiments (Table A.3) show that the model can produce higher-quality outputs in sampling mode, improving scores on CoNLL-2014-test benchmark, highlighting the value of this feature. In these experiments, we use a single model to generate several hypotheses in sampling mode, allowing us to select the chosen and rejected ones using a scoring approach and align the model to generate the preferred output. With this in mind, let's discuss these two scoring approaches.

4.3.2 GRECO

The GRECO model (Qorib and Ng, 2023) is the quality estimation model for GEC capable of identifying which words are correct or incorrect and also recognizing where additional words or phrases need to be inserted. It employs a BERT-like architecture to perform sequence classification considering concatenated source text and corrected hypothesis.

The quality score from the GRECO model is independent of the system generating the hypothesis and is based only on the text itself. The authors incorporated additional data in a system combination approach to achieve state-of-the-art results on the GEC task. This includes considering the number of systems that suggest a particular edit and the identities of these systems to generate the most accurate combined hypothesis of grammatically correct text. In our research, we utilize these scores to rank the corrections, identifying the chosen with high score and the rejected with low score samples of the preference dataset.

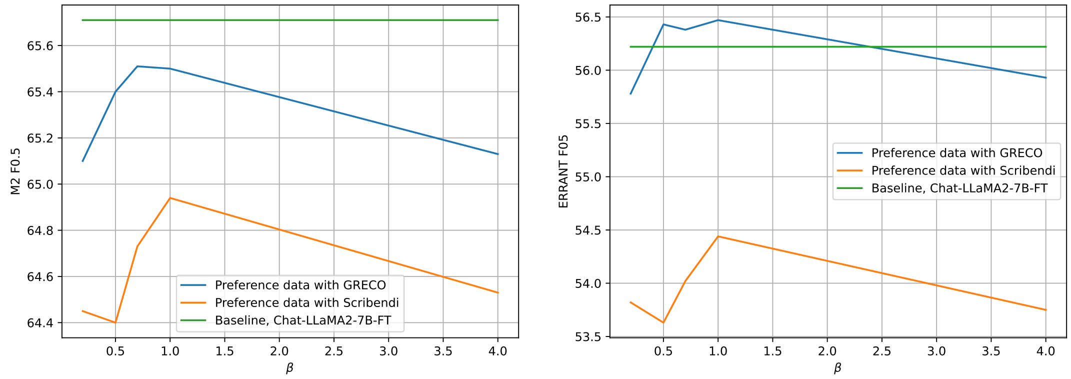
4.3.3 Scribendi score

The Scribendi Score, introduced in Islam and Magnani, 2021, evaluates the grammatical correctness of text using language models as a probabilistic method that calculates the word probability distribution in sequences within a corpus. These models are typically trained to minimize cross-entropy loss, effectively reducing perplexity, which has recently been used as a measure of writing quality (Islam and

Magnani, 2021). However, perplexity is an unbounded metric; thus, the Scribendi Score converts it into an absolute score (1=positive, -1=negative, 0=no change). This conversion uses a combination of language model perplexity (Radford et al., 2019) and sorted token/Levenshtein distance ratios to ensure that the corrected sentence is similar to the input and is more probable than the original.

Although these scores intuitively correlate with sentence grammatical correctness, they are not the most robust method for evaluating GEC systems, as the original approach used a pre-trained GPT-2 model without specific task conditioning. In this study, we believe this metric could be valuable for ranking hypotheses and creating a preference dataset for DPO.

4.3.4 Experiment results



(a) CoNLL-2014-test

(b) BEA-2019-dev

FIGURE 4.2: The dependency of $\mathbf{F_{0.5}}$ score on DPO hyper-parameter β for fine-tuned Chat-LLaMA2-7B-FT model with DPO approach on CoNLL-2014-test and BEA-2019-dev benchmarks.

In this experiment, we use the Chat-LLaMA-2-7B-FT model as both the reference model π_{ref} and the initial model for fine-tuning with DPO. We fine-tune the model using the PERF LoRA setting, updating $W_q, W_k, W_v, W_o, W_{gate}, W_{up}$ weights matrices, allowing us to fine-tune as much as possible weights. We used learning rate 10^{-5}, batch size of 2 with gradient accumulation 8. The final GEC quality measured on BEA-2019-dev and CoNLL-2014-test benchmarks for fine-tuned model in non-sampling mode with $temperature = 10^{-}3$ and $beam_{s}ize = 1$. We sourced sentences from the One Billion Word Benchmark corpus (Chelba et al., 2014) that was also used for Troy-1BW GEC dataset (Tarnavskyi, Chernodub, and Omelianchuk, 2022) and used a sampling strategy to generate 5 responses with grammatical corrections for each of the 20,000 sentences. Each response was evaluated by a scoring model to select pairs with the highest and lowest scores considered as chosen and rejected.

We adjusted the β hyper-parameter to control the penalty for deviations of the π_θ model from the reference model π_{ref}. Analysis of the $\mathbf{F_{0.5}}$ score dependency on β (Figure 4.2) shows that the GRECO Quality Estimation approach for creating a preference dataset outperforms the Scribendi score across the CoNLL-2014-test and BEA-2019-dev benchmarks. This advantage likely stems from the model hyper-parameters being specifically adapted to the open subsets CoNLL-2014-test and BEA-2019-dev, thus aligning more closely with the benchmarks' preferred edits. However, a higher β value, which penalizes the model for deviations from the reference model, unexpectedly resulted in lower scores.

Experiment	Training steps	CoNLL-2014-test			BEA-2019-dev		
		Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
All data	400	68.06	48.51	62.98	57.23	46.19	54.62
All data	800	67.92	49.26	63.14	57.21	46.86	54.79
Only TPs	400	67.73	47.87	62.54	57.65	46.82	55.1
Only TPs	800	68.29	49.14	**63.35**	57.22	46.8	54.78
Reference	-	68.42	46.24	62.43	57.35	45.93	54.63

TABLE 4.7: Compare DPO of Chat-LLaMA-2-7B-FT on preference data with sources from One Billion Word Benchmark corpus without filtering (all data) and on only pair with different chosen/rejected sentences.

The application of DPO for the GEC task yielded mixed results, showing a marginal increase of only 0.2 percentage points in the $\mathbf{F_{0.5}}$ score on the BEA-2019-dev, but a decrease in performance on the CoNLL-2014-test benchmarks when $\beta = 1$. Nonetheless, the metrics generally improved with increasing the number of training steps on both benchmarks (Table 4.7).

In our subsequent experiments (Table 4.7), we employed a preference dataset of 100,000 samples generated using the GRECO method from the One Billion Word Benchmark corpus. This dataset was organized into two configurations: one that included all data, resulting in 26% of the pairs (chosen/rejected) being identical, and a second, named "only TPs," which exclusively featured distinct chosen and rejected examples in each sample. Our results indicated that increasing the dataset size and using only distinct chosen and rejected pairs resulted in an improvement of 0.9% on the CoNLL-2014-test and 0.16% on the BEA-2019-dev. While these enhancements are not substantial, they indicate potential directions for further research.

We hypothesize that the limited improvement observed is due to the nature of the source data, which shares journalistic style and may not accurately reflect the error distribution found in benchmarks. Consequently, this limits our ability to provide a preference signal needed to improve model performance on the selected benchmarks. We employed the BEA-2019-train set to sample inputs using the Chat-LLaMA-2-7B-FT model to address this.

Furthermore, we suspect that the model's performance could not be significantly enhanced due to a bias introduced by our greedy chosen/rejected sampling strategy. In this strategy, the model generates samples in a sampling mode, making it highly likely that the rejected samples could never be produced by the model in a non-sampling generation mode. To mitigate this bias, we introduced a new dataset that includes the top-1 chosen and a randomly selected rejected sample from the remaining samples, aiming to diversify the preference data (Random rejected selection in Table 4.8).

Additionally, to improve the model's precision, we aimed to maintain the positive signal present in the original GEC dataset, W&I train set, and use all true negatives as chosen samples in the preference dataset as well as randomly selected rejected samples (Random rejected selection with TNs in Table 4.8).

From Table 4.8, the results demonstrate the overall positive impact of model alignment with DPO on the preference data sampled from W&I train set with randomly selected rejected samples, increasing the $\mathbf{F_{0.5}}$ scores on 0.3% and setting the absolute scores for CoNLL-2014-test to 66.05% and BEA-2019-dev to 56.54%. The increasing true negatives do not give us the expected precision improvement for CoNLL-2014-test.

Experiment	CoNLL-2014-test			BEA-2019-dev		
	Precision	**Recall**	$F_{0.5}$	**Precision**	**Recall**	$F_{0.5}$
Greedy chosen/rejected selection	70.15	**53.39**	66.01	58.54	47.53	55.95
Random rejected selection	**70.63**	52.48	**66.05**	59.63	**46.71**	56.51
Random rejected selection with TNs	70.48	51.95	65.78	**59.83**	46.35	**56.54**
Reference	70.28	52.12	65.71	59.43	46.23	56.22

TABLE 4.8: Compare DPO of Chat-LLaMA-2-7B-FT on preference data without filtering (all data) and on only pair with different positives/negatives (only TPs).

4.4 Conclusion

In this chapter, we explored the use of open-source Large Language Models in zero-shot and supervised fine-tuning settings for the Grammatical Error Correction task. We demonstrated that the correction quality of a model depends on its size and can be controlled with specific task formulations to perform precise corrections with minimal edits. Our findings show that the Instruct-Mistral-7B-ZS and Instruct-Gemma-7B-ZS models outperform the Chat-LLaMA-2-7B-ZS model in terms of recall, showcasing their ability to paraphrase and address the GEC task. However, improvements in precision and the $\mathbf{F_{0.5}}$ score remained challenging due to complexities in defining precise task formulations, establishing Chat-LLaMA-2-7B-ZS as the preferred choice for this task.

We established that supervised fine-tuning is necessary to develop a model that can compete with other open-source Grammatical Error Correction systems in a single-model setup. Utilizing a high-quality dataset such as W&I, we enhanced the model's error correction capabilities, achieving a performance that surpassed LLaMA-2-7B models trained on NUCLE and cLang-8 by 2 percentage points of $\mathbf{F_{0.5}}$ metric. The best-performing model was trained using a combination of all selected open-source datasets. However, we observed that no single-model approach consistently dominated across all benchmarks. Our experiments demonstrate the benefits of employing larger models and applying instruction-following tuning to improve GEC performance.

Although the fine-tuning method shows high-quality results, it is notably computationally intensive, requiring 30 times more time to complete a single training iteration. Consequently, we investigated the parameter-efficient tuning method, Low-Rank Adaptation (LoRA), as a less resource-intensive alternative to full fine-tuning. Our results showed that full model fine-tuning substantially outperformed the LoRA settings by 5 percentage points on both the CoNLL-2014-test and BEA-2019-dev benchmarks, indicating that the trade-off between extensive optimization and computational efficiency may not always be justified.

Finally, our limited set of experiments with model alignment through Direct Preference Optimization on preferred outputs selected by scoring metrics showed a modest improvement in the $\mathbf{F_{0.5}}$ score by 0.3%, indicating a further path for improvement for future work.

Chapter 5

Conclusions

5.1 Contribution

This thesis has explored the use of Large Language Models in the domain of Grammatical Error Correction, employing various approaches including zero-shot and supervised fine-tuning to improve the correction accuracy. Our research has provided substantial insights into several key areas:

1. Our investigations confirm that LLMs like Chat-LLaMA, Gemini, and Mistral are highly capable in GEC tasks. The zero-shot approach has shown promising results, highlighting the potential of LLMs to adapt to GEC tasks without extensive training. However, in terms of overall accuracy, zero-shot models still lag behind those that are fully fine-tuned, achieving 52.61% and 36.47% compared to 67.2% and 55.32% on the CoNLL-2014-test and BEA-2019-dev benchmarks respectively for Chat-LLaMA-7B-ZS.

2. We demonstrated that the formulation of tasks significantly impacts LLM performance. Adjusting prompts to include instructive elements can control the models' precision by up to 11%. Notably, models specifically designed to follow instructions, such as Instruct-Mistral-7B-ZS and Instruct-Gemma-7B-ZS, outperformed Chat-LLaMA-7B-ZS in terms of recall by 7%. These models show strong paraphrasing capabilities essential for addressing GEC, though improving precision and the $\mathbf{F_{0.5}}$ score remains a challenge due to the complexity of defining precise task formulations.

3. Our study compared full model weights fine-tuning and Parameter-Efficient Fine-Tuning strategies like Low-Rank Adaptation. LoRA fine-tuning offers a viable way to enhance model performance to 60.42% and 51.33% on CoNLL-2014-test and BEA-2019-dev benchmarks. However, increasing the fine-tuned parameter number is required to achieve the highest quality.

4. In full supervised fine-tuning settings, instruction tuning of larger models substantially improved the performance of the GEC system based on Chat-LLaMA-2-13B-FT, achieving 67.9% and 56.64% on the respective benchmarks. This performance is comparable to other state-of-the-art systems in a single-model setup, while using considerably less training data. Using a high-quality dataset such as W&I proved particularly effective, resulting in a model that outperformed LLaMA-2-7B models trained on NUCLE and cLang-8 by 2 percentage points $\mathbf{F_{0.5}}$ metric. The best results were achieved using a combination of all selected open-source datasets, although no single-model approach consistently dominated across all benchmarks.

5. We investigated the Preference Tuning approach, specifically Direct Preference Optimization to align model outputs with automatic metrics for GEC tasks. This involved the use of the GRECO model and Scribendi score for preference data generation. We found that incorporating distinct pairs of chosen/rejected sentences from the W&I train set and selecting randomly rejected text led to a modest improvement in the performance of the reference Chat-LLaMA-2-7B-FT model, achieving 66.05% and 56.51% of $\mathbf{F_{0.5}}$ metric on the CoNLL-2014-test and BEA-2019 dev benchmarks.

6. Our code and trained models are publicly available [1].

Finally, this thesis has significantly enhanced our understanding of LLMs' capabilities and optimization strategies in GEC tasks, laying a foundation for future research. The results underlines a strong potential for these models to transform writing assistance tools, making them more adaptable, efficient, and aligned with user expectations. Moving forward, continuing to refine these models and strategies will be essential in achieving the ultimate goal of developing highly accurate and user-friendly automated writing assistants.

5.2 Limitations

In this study, we focus only on the English language, potentially limiting the generalization of our findings to other languages. Additionally, we rely on evaluations using only two benchmarks and automatic metrics without incorporating human feedback to assess model quality. This restricts our ability to evaluate certain language nuances that might be better judged by humans.

We investigate only 2B, 7B, and 13B LLMs because these models do not have specific hardware requirements for both training and serving, making them suitable for production with reasonable costs.

Our study focuses solely on three foundational large language models: LLaMA-2, Mistral, and Gemma. We assume that the selected prompts and hyper-parameters originally adopted for the LLaMA-2 model are also applicable to the other models.

Our current work explores preference tuning with an artificial scoring function, allowing us to test the hypothesis of model alignment on the GEC task. This approach introduces bias to CoNLL-2014 and BEA-2019 benchmarks and does not incorporate human feedback, which is important in certain practical applications.

We investigate the models' ability to perform GEC tasks only in non-sampling mode for both supervised and preference fine-tuning settings. The quality of corrections generated in sampling mode was left out of the scope of this research and will be a topic for future work.

5.3 Future work

1. Develop and test automatic prompt optimization techniques to adapt models for GEC tasks, potentially enhancing the model's responsiveness to varied linguistic structures and error patterns.

2. Investigate the capabilities and performance of larger open-source LLMs, specifically those with around 70 billion parameters architectures, to understand how scale impacts quality in language tasks, especially GEC.

[1] https://github.com/ironiksk/gec-with-llms

3. Analyze how the rate of true negatives and the distribution of error types within training datasets affect model alignment with GEC benchmarks, aiming to improve precision and recall in model outputs in full supervised tuning setup.

4. Study the application of Direct Preference Optimization using preference data derived directly from the training set to refine model output quality and relevance.

5. Explore Preference tuning techniques such as Kahneman-Tversky Optimization and Contrastive Preference Optimization to as an alternative to DPO in GEC task.

6. Investigate the capabilities and performance of LLMs in languages other than English, such as Ukrainian, to assess and enhance multilingual error correction and language understanding capabilities.

Appendix A

Ablation study

Prompt ID	Prompt text
#6	*Fix all grammatical errors , do not rephrase .*
#14	*Fix all spelling, punctuation, grammar errors , do not rephrase .*
#15	*Fix all grammatical errors (spelling, punctuation, grammar) , do not rephrase .*
#16	*Fix all grammatical errors (spelling, punctuation) , do not rephrase .*
#17	*Fix all grammatical errors (spelling) , do not rephrase .*
#18	*Fix all grammatical errors (punctuation) , do not rephrase .*

TABLE A.1: GEC Task formulation for Zero-Shot setting with specific error types.

		CoNLL-2014-test			BEA-dev		
System	Prompt ID	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
Chat-LLaMa-2-7B-ZS	1	50.85	50.03	50.68	32.57	41.08	33.98
Chat-LLaMa-2-7B-ZS	2	42.17	54.76	44.2	22.4	40.99	24.63
Chat-LLaMa-2-7B-ZS	3	46.81	52.03	47.77	28.33	41.56	30.26
Chat-LLaMa-2-7B-ZS	4	50.94	50.54	50.86	32.38	41.36	33.85
Chat-LLaMa-2-7B-ZS	5	44.39	51.76	45.69	25.68	40.34	27.7
Chat-LLaMa-2-7B-ZS	6	53.62	48.39	52.61	35.64	40.22	36.47
Chat-LLaMa-2-7B-ZS	7	51.40	50.27	51.17	33.01	40.88	34.33
Chat-LLaMa-2-7B-ZS	8	50.03	50.20	50	31.78	41.62	33.36
Chat-LLaMa-2-7B-ZS	9	45.16	52.52	46.46	26.48	41.9	28.58
Chat-LLaMa-2-7B-ZS	10	51.62	49.77	51.24	32.94	40.89	34.27
Chat-LLaMa-2-7B-ZS	11	51.17	50.80	51.1	31.82	41.43	33.37
Chat-LLaMa-2-7B-ZS	12	50.44	51.14	50.58	30.76	40.5	32.32
Chat-LLaMa-2-7B-ZS	13	51.19	50.71	51.1	32.06	40.28	33.34
Chat-LLaMa-2-7B-ZS	14	54.19	46.52	52.46	36.65	38.45	37
Chat-LLaMa-2-7B-ZS	15	53.39	48.28	52.28	35.73	39.62	36.44
Chat-LLaMa-2-7B-ZS	16	53.91	**48.48**	52.72	35.96	**39.69**	36.65
Chat-LLaMa-2-7B-ZS	17	54.19	47.84	52.79	36.56	39.65	37.14
Chat-LLaMa-2-7B-ZS	18	**54.77**	47.84	**53.23**	**36.86**	39.14	**37.3**
Chat-LLaMa-2-13B	1	48.66	56.08	49.98			
Chat-LLaMa-2-13B	2	42.34	57.50	44.70			
Chat-LLaMa-2-13B	3	48.11	56.29	49.55			
Chat-LLaMa-2-13B	4	50.42	55.30	51.32			
Chat-LLaMa-2-13B	5	41.95	56.79	44.26			
Chat-LLaMa-2-13B	6	51.12	54.47	51.76	32.47	44.14	34.28
Chat-LLaMa-2-13B	7	51.35	53.37	51.74			
Chat-LLaMa-2-13B	8	50.53	54.57	51.29			
Chat-LLaMa-2-13B	12	46.47	56.45	48.18	26.68	44.83	29.03
Chat-LLaMa-2-13B	13	47.07	55.69	48.58			
Instruct-Mistral-7B-v0.2	1	39.60	56.10	42.09			
Instruct-Mistral-7B-v0.2	2	32.40	55.70	35.37			
Instruct-Mistral-7B-v0.2	3	36.39	57.59	39.29			
Instruct-Mistral-7B-v0.2	4	40.21	56.89	42.71			
Instruct-Mistral-7B-v0.2	5	30.93	54.94	33.89			
Instruct-Mistral-7B-v0.2	6	40.44	56.02	42.83	21.29	31.66	22.78
Instruct-Mistral-7B-v0.2	7	43.21	53.33	44.9	21.94	31.89	23.4
Instruct-Mistral-7B-v0.2	8	40.08	53.33	42.1			
Instruct-Mistral-7B-v0.2	9	34.86	55.15	37.63	12.68	29.02	13.29
Instruct-Mistral-7B-v0.2	10	39.89	53.83	42.07			
Instruct-Mistral-7B-v0.2	11	39.81	55.36	42.18			
Instruct-Mistral-7B-v0.2	12	37.45	55.24	40.03			
Instruct-Mistral-7B-v0.2	13	38.51	55.28	41.			
Instruct-Gemma-2B	1	43.03	26.52	38.27			
Instruct-Gemma-2B	2	39.69	38.84	39.52			
Instruct-Gemma-2B	3	43.36	34.36	41.2			
Instruct-Gemma-2B	4	47.61	33.57	43.93			
Instruct-Gemma-2B	5	39.05	37.09	38.64			
Instruct-Gemma-2B	6	40.03	37.36	39.46			
Instruct-Gemma-2B	7	44.98	32.73	41.84			
Instruct-Gemma-2B	8	40.5	33.21	38.82			
Instruct-Gemma-7B	6	50.3	53.32	50.87	31.44	42.92	33.22
Instruct-Gemma-2B	6	35.39	45.55	37.04	15.35	29.65	16.99

TABLE A.2: Comparison of the zero-shot setting on CoNLL-2014-test, BEA-dev.

	CoNLL-2014-test			BEA-dev		
temperature	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
0.001	74.48	49.59	67.68	56.34	47.02	54.19
0.7	74.68	49.57	67.67	56.37	47.02	54.21
1.0	74.48	49.64	67.71	56.33	47.02	54.18
2.0	74.55	49.68	67.77	56.33	47.03	54.19
Chat-LLaMA-2-7B-FT	75.40	46.84	67.20	58.26	46.03	55.32

TABLE A.3: Comparison of Chat-LLaMA2-7B-FT model quality in sampling mode with best candidate selected by GRECO depending on temperature on CoNLL-2014-test and BEA-dev.

	CoNLL-2014-test			**BEA-dev**		
β	**Precision**	**Recall**	$\mathbf{F_{0.5}}$	**Precision**	**Recall**	$\mathbf{F_{0.5}}$
0.2	68.56	54.15	65.1	57.82	48.88	55.78
0.5	69.23	53.53	65.4	58.75	48.75	56.43
0.7	69.53	53.2	65.51	58.77	48.52	56.38
1.0	69.57	53.08	65.5	58.48	48.33	56.47
4.0	68.61	54.16	65.13	57.97	49.01	55.93
Chat-LLaMA-2-7B-FT	70.28	52.12	65.71	59.43	46.23	56.22

TABLE A.4: Comparison of finetuned with DPO Chat-LLaMA-2-7B-FT model on GRECO-based preference data for Chat-LLaMA2-7B-FT on CoNLL-2014-test and BEA-dev. Learning rate $1 \cdot 10^{-5}$, number of updates 500, warm-up steps 20, batch size 8.

	CoNLL-2014-test			**BEA-dev**		
β	**Precision**	**Recall**	$\mathbf{F_{0.5}}$	**Precision**	**Recall**	$\mathbf{F_{0.5}}$
0.2	66.52	57.31	64.45	54.35	51.79	53.82
0.5	66.32	57.73	64.4	54.02	52.15	53.63
0.7	66.91	57.25	64.73	56.7	51.47	54.02
1.0	67.51	56.37	64.94	55.49	50.62	54.44
4.0	66.6	57.38	64.53	54.27	51.76	53.75
Chat-LLaMA-2-7B-FT	70.28	52.12	65.71	59.43	46.23	56.22

TABLE A.5: Comparison of finetuned with DPO Chat-LLaMA-2-7B-FT model on Scribendi-based preference data for Chat-LLaMA2-7B-FT on CoNLL-2014-test and BEA-dev. Learning rate $1 \cdot 10^{-5}$, number of updates 500, warm-up steps 20, batch size 8.

Trainable parameters name	Trainable parameters number	Computational time, sec/it
Full	6.7B	90
W_q, W_k, W_v, W_o	16.2M	3
W_q, W_v	4.2M	2.5

TABLE A.6: Comparison for time of 1 step weights update for Chat-LLaMA-7B-FT model on 4 Nvidia A10G GPUs with batch size 8.

Bibliography

Awasthi, Abhijeet et al. (Nov. 2019). "Parallel Iterative Edit Models for Local Sequence Transduction". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 4260–4270. DOI: 10.18653/v1/D19-1435. URL: https://aclanthology.org/D19-1435.

Bard, Gregory V. (2006). *Spelling-Error Tolerant, Order-Independent Pass-Phrases via the Damerau-Levenshtein String-Edit Distance Metric*. Cryptology ePrint Archive, Paper 2006/364. https://eprint.iacr.org/2006/364. URL: https://eprint.iacr.org/2006/364.

Brockett, Chris, William B. Dolan, and Michael Gamon (July 2006). "Correcting ESL Errors Using Phrasal SMT Techniques". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Ed. by Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle. Sydney, Australia: Association for Computational Linguistics, pp. 249–256. DOI: 10.3115/1220175.1220207. URL: https://aclanthology.org/P06-1032.

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL].

Bryant, Christopher, Mariano Felice, and Ted Briscoe (July 2017). "Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Regina Barzilay and Min-Yen Kan. Vancouver, Canada: Association for Computational Linguistics, pp. 793–805. DOI: 10.18653/v1/P17-1074. URL: https://aclanthology.org/P17-1074.

Bryant, Christopher and Hwee Tou Ng (July 2015). "How Far are We from Fully Automatic High Quality Grammatical Error Correction?" In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong and Michael Strube. Beijing, China: Association for Computational Linguistics, pp. 697–707. DOI: 10.3115/v1/P15-1068. URL: https://aclanthology.org/P15-1068.

Bryant, Christopher et al. (Aug. 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*. Ed. by Helen Yannakoudakis et al. Florence, Italy: Association for Computational Linguistics, pp. 52–75. DOI: 10.18653/v1/W19-4406. URL: https://aclanthology.org/W19-4406.

Bryant, Christopher et al. (July 2023). "Grammatical Error Correction: A Survey of the State of the Art". In: *Computational Linguistics*, 1–59. ISSN: 1530-9312. DOI: 10.1162/coli_a_00478. URL: http://dx.doi.org/10.1162/coli_a_00478.

Chelba, Ciprian et al. (2014). *One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling*. arXiv: 1312.3005 [cs.CL].

Chollampatt, Shamil and Hwee Tou Ng (Aug. 2018). "A Reassessment of Reference-Based Grammatical Error Correction Metrics". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Ed. by Emily M. Bender, Leon Derczynski, and Pierre Isabelle. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 2730–2741. URL: https://aclanthology.org/C18-1231.

Coyne, Steven and Keisuke Sakaguchi (2023). "An Analysis of GPT-3's Performance in Grammatical Error Correction". In: *ArXiv* abs/2303.14342. URL: https://api.semanticscholar.org/CorpusID:257766625.

Dahlmeier, Daniel and Hwee Tou Ng (June 2012). "Better Evaluation for Grammatical Error Correction". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Eric Fosler-Lussier, Ellen Riloff, and Srinivas Bangalore. Montréal, Canada: Association for Computational Linguistics, pp. 568–572. URL: https://aclanthology.org/N12-1067.

Dahlmeier, Daniel, Hwee Tou Ng, and Siew Mei Wu (June 2013). "Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English". In: *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Joel Tetreault, Jill Burstein, and Claudia Leacock. Atlanta, Georgia: Association for Computational Linguistics, pp. 22–31. URL: https://aclanthology.org/W13-1703.

Dale, Robert, Ilya Anisimoff, and George Narroway (June 2012). "HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task". In: *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Ed. by Joel Tetreault, Jill Burstein, and Claudia Leacock. Montréal, Canada: Association for Computational Linguistics, pp. 54–62. URL: https://aclanthology.org/W12-2006.

De Felice, Rachele and Stephen G. Pulman (Aug. 2008). "A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English". In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Ed. by Donia Scott and Hans Uszkoreit. Manchester, UK: Coling 2008 Organizing Committee, pp. 169–176. URL: https://aclanthology.org/C08-1022.

Ding, Ning et al. (2022). *Delta Tuning: A Comprehensive Study of Parameter Efficient Methods for Pre-trained Language Models*. arXiv: 2203.06904 [cs.CL].

Fang, Tao et al. (2023). *Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation*. arXiv: 2304.01746 [cs.CL].

Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Edward Gillian (Jan. 2015). "Human Evaluation of Grammatical Error Correction Systems". In: pp. 461–470. DOI: 10.18653/v1/D15-1052.

Hotate, Kengo, Masahiro Kaneko, and Mamoru Komachi (2020). "Generating Diverse Corrections with Local Beam Search for Grammatical Error Correction". In: *International Conference on Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:227231718.

Houlsby, Neil et al. (2019). *Parameter-Efficient Transfer Learning for NLP*. arXiv: 1902.00751 [cs.LG].

Hu, Edward J. et al. (2021). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv: 2106.09685 [cs.CL].

Islam, Md Asadul and Enrico Magnani (Nov. 2021). "Is this the end of the gold standard? A straightforward reference-less grammatical error correction metric". In:

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 3009–3015. DOI: 10.18653/v1/2021.emnlp-main.239. URL: https://aclanthology.org/2021.emnlp-main.239.

Jensen, K. et al. (1983). "Parse Fitting and Prose Fixing: Getting a Hold on Ill-Formedness". In: *American Journal of Computational Linguistics* 9.3-4, pp. 147–160. URL: https://aclanthology.org/J83-3002.

Jiang, Albert Q. et al. (2023). *Mistral 7B*. arXiv: 2310.06825 [cs.CL].

Junczys-Dowmunt, Marcin et al. (June 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)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, pp. 595–606. DOI: 10.18653/v1/N18-1055. URL: https://aclanthology.org/N18-1055.

Kaneko, Masahiro and Naoaki Okazaki (2023). *Reducing Sequence Length by Predicting Edit Operations with Large Language Models*. arXiv: 2305.11862 [cs.CL].

Kaneko, Masahiro et al. (July 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*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, pp. 4248–4254. DOI: 10.18653/v1/2020.acl-main.391. URL: https://aclanthology.org/2020.acl-main.391.

Kwasny, Stan C. and Norman K. Sondheimer (1981). "Relaxation Techniques for Parsing Grammatically Ill-Formed Input in Natural Language Understanding Systems". In: *American Journal of Computational Linguistics* 7.2, pp. 99–108. URL: https://aclanthology.org/J81-2002.

Leacock, Claudia et al. (Jan. 2010). *Automated Grammatical Error Detection for Language Learners, Second Edition*. Vol. 7. DOI: 10.2200/S00275ED1V01Y201006HLT009.

Li, Xiang Lisa and Percy Liang (2021). *Prefix-Tuning: Optimizing Continuous Prompts for Generation*. arXiv: 2101.00190 [cs.CL].

Loem, Mengsay et al. (2023). *Exploring Effectiveness of GPT-3 in Grammatical Error Correction: A Study on Performance and Controllability in Prompt-Based Methods*. arXiv: 2305.18156 [cs.CL].

Malmi, Eric et al. (Nov. 2019). "Encode, Tag, Realize: High-Precision Text Editing". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by Kentaro Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 5054–5065. DOI: 10.18653/v1/D19-1510. URL: https://aclanthology.org/D19-1510.

Mita, Masato and Hitomi Yanaka (2021). *Do Grammatical Error Correction Models Realize Grammatical Generalization?* arXiv: 2106.03031 [cs.CL].

Mizumoto, Tomoya et al. (Dec. 2012). "The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings". In: *Proceedings of COLING 2012: Posters*. Ed. by Martin Kay and Christian Boitet. Mumbai, India: The COLING 2012 Organizing Committee, pp. 863–872. URL: https://aclanthology.org/C12-2084.

Napoles, Courtney, Maria Nădejde, and Joel Tetreault (2019). "Enabling Robust Grammatical Error Correction in New Domains: Data Sets, Metrics, and Analyses". In: *Transactions of the Association for Computational Linguistics* 7. Ed. by Lillian Lee et al., pp. 551–566. DOI: 10.1162/tacl_a_00282. URL: https://aclanthology.org/Q19-1032.

Napoles, Courtney, Keisuke Sakaguchi, and Joel Tetreault (Apr. 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*. Ed. by Mirella Lapata, Phil Blunsom, and Alexander Koller. Valencia, Spain: Association for Computational Linguistics, pp. 229–234. URL: https://aclanthology.org/E17-2037.

Ng, Hwee Tou et al. (Aug. 2013). "The CoNLL-2013 Shared Task on Grammatical Error Correction". In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Ed. by Hwee Tou Ng et al. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1–12. URL: https://aclanthology.org/W13-3601.

Ng, Hwee Tou et al. (June 2014). "The CoNLL-2014 Shared Task on Grammatical Error Correction". In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Ed. by Hwee Tou Ng et al. Baltimore, Maryland: Association for Computational Linguistics, pp. 1–14. DOI: 10.3115/v1/W14-1701. URL: https://aclanthology.org/W14-1701.

Nicholls, Diane (1999). "The Cambridge Learner Corpus-Error coding and analysis". In: URL: https://api.semanticscholar.org/CorpusID:15295088.

Omelianchuk, Kostiantyn et al. (July 2020). "GECToR – Grammatical Error Correction: Tag, Not Rewrite". In: *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Jill Burstein et al. Seattle, WA, USA → Online: Association for Computational Linguistics, pp. 163–170. DOI: 10.18653/v1/2020.bea-1.16. URL: https://aclanthology.org/2020.bea-1.16.

Omelianchuk, Kostiantyn et al. (2024). *Pillars of Grammatical Error Correction: Comprehensive Inspection Of Contemporary Approaches In The Era of Large Language Models*. arXiv: 2404.14914 [cs.CL].

Ouyang, Long et al. (2022). *Training language models to follow instructions with human feedback*. arXiv: 2203.02155 [cs.CL].

Qorib, Muhammad Reza, Seung-Hoon Na, and Hwee Tou Ng (July 2022). "Frustratingly Easy System Combination for Grammatical Error Correction". In: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz. Seattle, United States: Association for Computational Linguistics, pp. 1964–1974. DOI: 10.18653/v1/2022.naacl-main.143. URL: https://aclanthology.org/2022.naacl-main.143.

Qorib, Muhammad Reza and Hwee Tou Ng (Dec. 2023). "System Combination via Quality Estimation for Grammatical Error Correction". In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Ed. by Houda Bouamor, Juan Pino, and Kalika Bali. Singapore: Association for Computational Linguistics, pp. 12746–12759. DOI: 10.18653/v1/2023.emnlp-main.785. URL: https://aclanthology.org/2023.emnlp-main.785.

Radford, Alec et al. (2019). "Language Models are Unsupervised Multitask Learners". In: URL: https://api.semanticscholar.org/CorpusID:160025533.

Rafailov, Rafael et al. (2023). *Direct Preference Optimization: Your Language Model is Secretly a Reward Model*. arXiv: 2305.18290 [cs.LG].

Rothe, Sascha et al. (Aug. 2021). "A Simple Recipe for Multilingual Grammatical Error Correction". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Ed. by Chengqing Zong et al. Online:

Association for Computational Linguistics, pp. 702–707. DOI: 10.18653/v1/2021.acl-short.89. URL: https://aclanthology.org/2021.acl-short.89.

Rozovskaya, Alla and Dan Roth (June 2010). “Training Paradigms for Correcting Errors in Grammar and Usage”. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Ed. by Ron Kaplan et al. Los Angeles, California: Association for Computational Linguistics, pp. 154–162. URL: https://aclanthology.org/N10-1018.

Sakaguchi, Keisuke et al. (2016). “Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality”. In: *Transactions of the Association for Computational Linguistics* 4. Ed. by Lillian Lee, Mark Johnson, and Kristina Toutanova, pp. 169–182. DOI: 10.1162/tacl_a_00091. URL: https://aclanthology.org/Q16-1013.

Schler, Jonathan et al. (2006). “Effects of Age and Gender on Blogging”. In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*. URL: https://api.semanticscholar.org/CorpusID:2075411.

Schulman, John et al. (2017). *Proximal Policy Optimization Algorithms*. arXiv: 1707.06347 [cs.LG].

Stahlberg, Felix and Shankar Kumar (Nov. 2020). “Seq2Edits: Sequence Transduction Using Span-level Edit Operations”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, pp. 5147–5159. DOI: 10.18653/v1/2020.emnlp-main.418. URL: https://aclanthology.org/2020.emnlp-main.418.

— (Apr. 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*. Ed. by Jill Burstein et al. Online: Association for Computational Linguistics, pp. 37–47. URL: https://aclanthology.org/2021.bea-1.4.

Susanto, Raymond Hendy, Peter Phandi, and Hwee Tou Ng (Oct. 2014). “System Combination for Grammatical Error Correction”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. Doha, Qatar: Association for Computational Linguistics, pp. 951–962. DOI: 10.3115/v1/D14-1102. URL: https://aclanthology.org/D14-1102.

Tajiri, Toshikazu, Mamoru Komachi, and Yuji Matsumoto (July 2012). “Tense and Aspect Error Correction for ESL Learners Using Global Context”. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Haizhou Li et al. Jeju Island, Korea: Association for Computational Linguistics, pp. 198–202. URL: https://aclanthology.org/P12-2039.

Tarnavskyi, Maksym, Artem Chernodub, and Kostiantyn Omelianchuk (May 2022). “Ensembling and Knowledge Distilling of Large Sequence Taggers for Grammatical Error Correction”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Ed. by Smaranda Muresan, Preslav Nakov, and Aline Villavicencio. Dublin, Ireland: Association for Computational Linguistics, pp. 3842–3852. DOI: 10.18653/v1/2022.acl-long.266. URL: https://aclanthology.org/2022.acl-long.266.

Team, Gemma et al. (2024). *Gemma: Open Models Based on Gemini Research and Technology*. arXiv: 2403.08295 [cs.CL].

Tetreault, Joel, Jennifer Foster, and Martin Chodorow (July 2010). “Using Parse Features for Preposition Selection and Error Detection”. In: *Proceedings of the ACL 2010 Conference Short Papers*. Ed. by Jan Hajič et al. Uppsala, Sweden: Association for Computational Linguistics, pp. 353–358. URL: https://aclanthology.org/P10-2065.

Touvron, Hugo et al. (2023). "Llama 2: Open foundation and fine-tuned chat models". In: *arXiv preprint arXiv:2307.09288*.

Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Wan, Zhaohong, Xiaojun Wan, and Wenguang Wang (Dec. 2020). "Improving Grammatical Error Correction with Data Augmentation by Editing Latent Representation". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 2202–2212. DOI: 10.18653/v1/2020.coling-main.200. URL: https://aclanthology.org/2020.coling-main.200.

Wei, Jason et al. (2021). *Frequency Effects on Syntactic Rule Learning in Transformers*. arXiv: 2109.07020 [cs.CL].

Xue, Linting et al. (2021). *mT5: A massively multilingual pre-trained text-to-text transformer*. arXiv: 2010.11934 [cs.CL].

Yannakoudakis, Helen, Ted Briscoe, and Ben Medlock (June 2011). "A New Dataset and Method for Automatically Grading ESOL Texts". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, pp. 180–189. URL: https://aclanthology.org/P11-1019.

Yuan, Zheng, Ted Briscoe, and Mariano Felice (June 2016). "Candidate re-ranking for SMT-based grammatical error correction". In: *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Joel Tetreault et al. San Diego, CA: Association for Computational Linguistics, pp. 256–266. DOI: 10.18653/v1/W16-0530. URL: https://aclanthology.org/W16-0530.

Yuan, Zheng et al. (Nov. 2021). "Multi-Class Grammatical Error Detection for Correction: A Tale of Two Systems". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, pp. 8722–8736. DOI: 10.18653/v1/2021.emnlp-main.687. URL: https://aclanthology.org/2021.emnlp-main.687.

Zhang, Yue et al. (2023). *Multi-Task Instruction Tuning of LLaMa for Specific Scenarios: A Preliminary Study on Writing Assistance*. arXiv: 2305.13225 [cs.CL].