applied sciences

MDPI

Article

Towards Harnessing the Most of ChatGPT for Korean Grammatical Error Correction

Chanjun Park [1,†], Seonmin Koo [2,†], Gyeongmin Kim [3] and Heuiseok Lim [2,3,*]

[1] Upstage, 338, Gwanggyojungang-ro, Suji-gu, Yongin-si 16942, Gyeonggi-do, Republic of Korea; chanjun.park@upstage.ai
[2] Department of Computer Science and Engineering, Korea University, 145, Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea; fhdahd@korea.ac.kr
[3] Human-Inspired AI Research, 145, Anam-ro, Seongbuk-gu, Seoul 02841, Republic of Korea; totoro4007@gmail.com
* Correspondence: limhseok@korea.ac.kr
† These authors contributed equally to this work.

Abstract: In this study, we conduct a pioneering and comprehensive examination of ChatGPT's (GPT-3.5 Turbo) capabilities within the realm of Korean Grammatical Error Correction (K-GEC). Given the Korean language's agglutinative nature and its rich linguistic intricacies, the task of accurately correcting errors while preserving Korean-specific sentiments is notably challenging. Utilizing a systematic categorization of Korean grammatical errors, we delve into a meticulous, case-specific analysis to identify the strengths and limitations of a ChatGPT-based correction system. We also critically assess influential parameters like temperature and specific error criteria, illuminating potential strategies to enhance ChatGPT's efficacy in K-GEC tasks. Our findings offer valuable contributions to the expanding domain of NLP research centered on the Korean language.

Keywords: Korean grammatical error correction; large language model; K-NCT; ChatGPT

Citation: Park, C.; Koo, S.; Kim, G.; Lim, H. Towards Harnessing the Most of ChatGPT for Korean Grammatical Error Correction. *Appl. Sci.* **2024**, *14*, 3195. https://doi.org/10.3390/app14083195

Academic Editor: Stefan Fischer

Received: 19 March 2024
Revised: 8 April 2024
Accepted: 9 April 2024
Published: 10 April 2024

1. Introduction

Grammatical error correction systems identify and rectify errors in sentences. The challenge is pronounced in Korean due to its morphological richness and agglutinative nature [1–3]. The primary method for Korean error correction is rule-based, where specific error types are mapped to correction rules [4]. While effective in preserving sentence structure, this approach has its pitfalls: it is labor-intensive and rigid, failing to catch errors outside predefined rules. To mitigate this, statistical approaches have been introduced, basing error judgments on corpus-derived probabilities [5,6]. However, these require substantial corpora to achieve satisfactory performance.

Deep learning has recently emerged as a potent tool for grammatical correction, notably overcoming the challenges of prior approaches. Methods have been introduced specifically for Korean, enabling error correction model training without parallel data [1,7,8]. These primarily rely on noise injection processes that yield pseudo-parallel corpora from unlabeled mono corpora [1,9]. Typically, a sentence from the mono corpus serves as the target, with the noised variant acting as the source for training sequence-to-sequence models.

With the advent of large language models (LLMs) like ChatGPT [10], tasks previously researched in parallel in the NLP domain now converge under a single model. The model's capabilities emerge based on the input prompt, aligning outputs to user objectives [11–15].

In NLP, a "prompt" refers to the input query or statement. "Prompt engineering" focuses on designing and leveraging these prompts effectively. With this technique, one can precisely define the input prompt and provide clear task-specific information, enabling the model to yield more accurate outputs. Prompt engineering is actively employed in prominent LLMs like ChatGPT and GPT-4 [16]. From an LLM perspective, prompts are

user-defined commands for desired outcomes, and prompt engineering aims to identify optimal prompts to enhance output quality. Current research focuses on devising the best prompts to harness LLM's inherent capabilities. This includes studies on adversarial prompting, adversarial prompt detectors, prompt managers like TaskMatrix.AI [17], and parameter-efficient fine-tuning (PEFT) approaches such as P-Tuning and LoRA [18].

Despite advances in deep learning and LLM-based methods, two critical challenges persist. First, the lack of a standardized grammatical error correction dataset results in inconsistent model evaluations. Researchers often resort to creating test sets from arbitrary corpus samples, leading to non-uniform benchmarks across studies, potentially skewing performance metrics [19,20]. Second, the variability in error types defined across research further compromises assessment reliability. Without a precise error typology standard, a model's efficacy might be misjudged based on the specific test set used, either under or overestimating its performance [2].

Therefore, Refs. [21–23] highlighted the shortcomings in data creation and evaluation in prior Korean grammatical correction research. They introduced an error-type classification system and, leveraging this, constructed the inaugural gold-standard test set for Korean Grammatical Error Correction (K-GEC), termed K-NCT.

In light of the recent advancements and shifts in the field, this study aims to embark on a comprehensive exploration and validation of the performance of the K-GEC task, specifically grounded on the K-NCT test set, using the ChatGPT framework. We endeavor to critically analyze, benchmark, and interpret the results within the broader context of the evolution of NLP methodologies. Our goal is not only to measure raw performance metrics but also to understand the intricate nuances, potential challenges, and areas of improvement that the K-NCT dataset might present when paired with advanced models like ChatGPT.

In light of the recent advancements and shifts in the field, this study aims to embark on a comprehensive exploration and validation of the performance of the K-GEC task, specifically grounded on the K-NCT test set, using the ChatGPT framework. We endeavor to critically analyze, benchmark, and interpret the results within the broader context of the evolution of NLP methodologies. Our goal is not only to measure raw performance metrics but also to understand the intricate nuances, potential challenges, and areas of improvement that the K-NCT dataset might present when paired with advanced models like ChatGPT. Our main contributions are threefold:

- Comprehensive evaluation of the performance of the state-of-the-art large language model ChatGPT on the Korean Grammatical Error Correction (K-GEC) task using the K-NCT gold-standard test set.
- Critical interpretation of the results within the broader context of the evolution of NLP methodologies, examining the potential challenges, strengths, and areas for improvement that the K-NCT dataset might unveil when used in conjunction with cutting-edge language models like ChatGPT.
- Exploration of the role of prompt engineering in optimizing ChatGPT's performance on the K-GEC task, investigating how carefully crafted prompts can influence the model's outputs and accuracy.

The paper is divided as follows. Section 3 describes the types of errors and validation design for assessing the performance of ChatGPT on the K-GEC task. Section 4 delineates the experimental settings utilizing the K-NCT dataset. Section 5.1 expounds on the experimental results concerning various temperatures and shots. Sections 5.2 and 5.3 provide detailed interpretations through additional analysis and qualitative analysis, respectively.

2. Related Works

In this section, we introduce related works on the GEC task and LLMs. This aids in understanding the research aimed at harnessing the capabilities of LLMs for the Korean GEC that we are undertaking.

2.1. Korean Grammatical Error Correction

Deep learning paradigms for Grammatical Error Correction (GEC) often treat the problem akin to machine translation: the objective is to "translate" an erroneous sentence into its grammatically correct counterpart. This is predominantly achieved through sequence-to-sequence frameworks, leveraging noising encoders and denoising decoders [24–26].

There has been a noticeable surge in GEC research tailored to high-resource languages. Novel approaches have emerged, such as the sequence tagging models built on transformer architectures for both error detection and rectification [27]. Educative interfaces, aiming to assist language learners, now encompass enhanced interpretability with explicative texts and examples delineating reasons behind each correction [28]. Techniques like Self-Supervised Curriculum Learning gauge data complexity via training losses, optimizing models for better performance [29]. Moreover, contrastive learning methodologies show promise, especially for domains characterized by sparse errors [30]. On the efficiency frontier, aggressive decoding has been shown to boost model speed by rapidly decoding a multitude of tokens [31].

However, the application of these advanced methods often hinges on the availability of parallel corpora, encapsulating pairs of incorrect and correct sentences. A significant hurdle for the Korean language is the absence of such publicly accessible datasets. To circumvent this, research efforts in low-resource, including Korean, are concentrated on creating pseudo-parallel corpora autonomously, eliminating the need for manual intervention. In the Chinese GEC task, automatic noise insertion is conducted by incorporating visually or phonologically resembled characters to generate erroneous sentences [32]. Additionally, Ref. [33] focuses on the Swedish GEC task. The cornerstone of this approach is the automatic noise generation technique, wherein designed noise functions are imposed on mono corpora to generate pseudo parallel sets [34]. Noise is typically introduced in the form of grapheme-to-phoneme deviations, misplacements of spacing and punctuation, and pronunciation errors, among others. Nonetheless, a consistent challenge remains: different studies employ varied error definitions, and when training and testing datasets are derived from pseudo-generated data, there's a propensity for overlapping error types. This overlap can compromise the objectivity of evaluations, making unbiased research a daunting task.

Notably, while there's been a surge of advancements in various methodologies for GEC, a significant gap remains: the exploration of LLMs within this domain. LLMs, with their expansive parameter sets and generalized training, have transformed various segments of the NLP landscape. To assess the potential utilization of LLMs in the GEC task, research has been conducted across various languages such as English, Chinese, Japanese, and others [35–37]. Yet, their application to K-GEC remains largely uncharted territory. To the best of our knowledge and based on our literature review, there has not been a concerted effort to harness the capabilities of LLMs in Korean, such as ChatGPT [10] or GPT-4 [16], for this task. This paper seeks to bridge this gap, venturing into the application and analysis of LLMs for K-GEC, probing their potential in redefining error correction mechanisms for morphologically rich languages like Korean.

2.2. Large Language Model

The emergence of Large Language Models (LLMs) like ChatGPT [10] signals a paradigm shift in the NLP realm. Previously distinct tasks such as translation, summarization, question-answering, and morphological analysis now converge under the umbrella of a single model, influenced significantly by the input prompt [38]. This convergence paradigm hinges on the careful design of prompts, which unlock the multifaceted capabilities of LLMs to cater to diverse user objectives.

Recent advancements in LLMs can be attributed to three pivotal factors. First, the digital revolution has flooded the research domain with an unprecedented volume of text data. This abundance, a keystone in modern research, allows LLMs to harness extensive corpora, amplifying their generalization prowess and facilitating intricate learning across diverse contexts and topics. Second, strides in computing, underscored by the

emergence of high-throughput parallel-processing units such as GPUs and TPUs, have substantially alleviated computational constraints. This technological surge enables researchers to devise increasingly complex and deep neural architectures. Lastly, the introduction of attention mechanisms and the transformative transformer architecture has markedly redefined context modeling in NLP, paving the way for refined representations of context interrelations [39,40].

Central to these shifts is the notion of 'scaling laws' [41], which posits a positive correlation between model size and performance. Consequently, increasing model parameters aligns with systematic explorations of performance boosts. The interplay between academic challenges and technological advances fuels LLM advancements, positioning them as pivotal in shaping future NLP research trajectories.

3. ChatGPT for K-GEC

GEC tasks necessitate a comprehension of sentence structures and grammar, coupled with the ability to generate suitable corrective words. The outcomes of grammatical corrections can influence various downstream tasks, potentially impacting both model performance and user satisfaction. In light of this significance, this study aims to evaluate the performance of ChatGPT, an LLM, on the K-GEC task. For the evaluation, we will meticulously categorize and assess grammatical errors manifested in sentences.

3.1. Types of Validation

To analyze the proficiency of LLMs in the K-GEC task, a holistic metric is not sufficient; an in-depth analysis per error type is imperative. Therefore, we evaluate across a total of 21 error categories. Table 1 shows descriptions of the categories under scrutiny. **Spacing Error**: Breaching Korean spacing standards can happen, and just like in deep learning models, such mistakes might arise from individuals typing rapidly or from certain habits. **Punctuation Error**: Incorrect punctuation usage can alter the meaning of a sentence. This mistake might be made by deep learning models, particularly when confronted with unfamiliar terms. **Numerical Error**: There can be a mix-up between cardinal and ordinal numbers. For example, '두 번째 (The second)' could be mistakenly typed as '둘 번째 (The two)'.

Breaching Korean grammar and spelling standards is commonplace. Such mistakes are categorized into primary and secondary, which can occur concurrently. The primary error is as follows: **Remove Error**: Certain words or their endings may be omitted, resulting in this error type. Such omissions are a recognized error category in Korean. **Addition Error**: This includes the unnecessary repetition of words, overlooking postpositions, or erroneously adding endings, which can stem from rapid typing or a lack of grammatical understanding. **Replace Error**: Errors like substituting one word with another or misarranging syllables within words often arise from fast typing. **Separation Error**: Separating the consonants and vowels within characters is frequent, often because the space key is typically employed for word separation. **Typing language Error**: These errors occur when typing while the keyboard mode is not set to Korean. For instance, '아침' might be typed as 'dkcla'. **Foreign word conversion Error**: Korean has its set standards for foreign word pronunciations. Deviations like writing '쌈바 (ssamba)' instead of '삼바 (samba)' result in errors.

The secondary error is as follows: **Consonant vowel conversion error**: Errors in non-verbal units can occur, like writing '툰은 통장에 넣어두세요. (Put your toons in a bank account)' instead of the correct '돈은 통장에 넣어두세요. (Put your money in a bank account)'. **Grapheme-to-phoneme(G2P) Error**: When spelling is based on pronunciation, mistakes like '나는 밥을 먹는다 (I eat rice)' instead of '나는 바블 먹는다 (I ate rice)' can happen. **Element Error**: Misarrangement or improper sequencing of Korean sentence components, which have a traditional structure, can lead to this error. **Tense Error**: Using an inappropriate verb tense, like a past tense verb in a future context. **Postposition Error**: Misuse of postpositions in Korean grammar. Given the Korean language's agglutinative nature, the proper use of verbs is crucial. **Suffix Error**: Misapplication of endings depending on

the context or situation. **Auxiliary predicate Error**: Misuse of auxiliary verbs, which are essential for constructing Korean sentences. **Dialect Error**: Utilizing non-standard regional language variations. Determining if something is an error depends on the intention of the speaker or writer. **Polite speech Error**: Using an inappropriately formal or informal expression, indicative of Korean cultural nuances. **Behavioral Error**: Assigning an action to a subject incapable of it, such as 'the banana eats the orange'. **Coreference Error**: Erroneous references in sentences can produce unintended outcomes. **Discourse context Error**: Producing sentences that are inconsistent with previous content or context. **Neologism Error**: Incorporating words or spellings that are not acknowledged in standard grammar. Determination of an error, similar to dialects, depends on the user's intent.

Table 1. Error type classification criteria for Korean grammatical and spelling error correction in K-NCT dataset [21].

Error Type					Explanation
Spacing Error					Violating the spacing rules
Punctuation Error					Punctuation marks are not attached in Korean sentences or are attached in the wrong position
Numerical Error					Cardinal number indicating quantity and the ordinal number indicating the order are in error
Spelling and Grammatical Error	Primary Error	Monolingual Error	Remove Error		Some words are not recognized, or endings or suffixes are omitted
			Addition Error		Same word is repeated, or an unused postposition or ending is added
			Replace Error	Word replace	Word is replaced by another word
				Rotation replace	Order of syllables changes within a one phrase
			Separation Error		Separating consonants and vowels in characters
		Multilingual Error	Typing language Error		Typing while the keyboard is not in Korean mode
			Foreign word conversion Error		Writing differently from the standard foreign language pronunciation
	Secondary Error	Spelling Error	Consonant vowel conversion error		Spelling error in non-speaking alphabet units
			Grapheme-to-phoneme(G2P) Error		Writing spellings according to pronunciation
		Syntax Error	Element Error		The Korean sentence components are not equipped or the word order is not correct
			Tense Error		Using a verb that does not match the tense
			Postposition Error		Probing that does not fit the grammar
			Suffix Error		Using an ending that is not grammatically correct
			Auxiliary predicate Error		Using an auxiliary verb that is not grammatically correct
		Semantic Error	Dialect Error		Writing in non-standard language
			Polite speech Error		An adjective expression that does not fit the subject
			Behavioral Error		Expressions that the subject cannot perform
			Coreference Error		Invalid entity reference
			Discourse context Error		Contradicting the context of the previous discourse
		Neologism Error			Using grammar or new words that are not included in the existing grammar system

3.2. Validation Design

The K-GEC task necessitates an understanding of sentence structure and grammar, as well as the capability to generate appropriate corrective words. The outcomes of grammatical correction can serve as inputs for various downstream tasks, potentially influencing the model's efficacy and user satisfaction. Given its significance, we aim to conduct a comprehensive evaluation of ChatGPT's performance on the K-GEC task, focusing on 21 error categories. For this analysis, we consider two dimensions of ChatGPT's capabilities: explicit example information and temperature.

The number of shots, or the provision of explicit examples for a given task, has been observed to have a significant impact on the performance of language models like ChatGPT. Specifically, providing a few examples (few-shot learning) has been shown to enhance the

model's ability to understand and execute tasks more effectively compared to a zero-shot setting, where no examples are provided.

While the majority of existing studies have focused on analyzing the performance of ChatGPT in a zero-shot setting [42,43], there is a lack of comprehensive analysis on the relative advantages and trade-offs associated with supplying the model with task-specific examples. The literature in this area remains limited, leaving a gap in our understanding of how providing examples can potentially improve the model's performance on various tasks.

It is widely acknowledged that few-shot learning can be beneficial for language models like ChatGPT, as it provides the model with a better understanding of the task at hand and the expected output format. By observing a small number of examples, the model can leverage its ability to learn patterns and adapt its behavior accordingly, potentially leading to improved task execution and more accurate or relevant responses.

Consequently, our research seeks to investigate the performance discrepancy between zero-shot and few-shots scenarios, and to evaluate the impact of varying explicit example counts on performance. We delve into the influence of shots across five settings (0, 1, 4, 8, 16). For error types within the typing language category, where the number of validation sentences is limited, we restrict our analysis to the performance differences between zero-shot and few-shots. To achieve this, we segregate examples, pertaining to the 21 error types we wish to validate, in advance to ensure they are distinct from the validation sentences we aim to examine.

Temperature is a crucial hyperparameter that significantly influences the nature of responses generated by language models like ChatGPT. It controls the randomness or uncertainty in the model's predictions during the decoding process. A higher temperature setting increases the probability of sampling from a broader range of the model's output distribution, leading to more diverse, unpredictable, and potentially creative responses. Conversely, a lower temperature setting narrows the model's focus, favoring more predictable, safer, and grammatically correct outputs.

In the context of the GEC task, we aim to investigate how varying the temperature parameter impacts the model's performance. Specifically, we intend to evaluate the model's GEC capabilities across three distinct temperature settings: 0.2 (low), 0.5 (medium), and 0.8 (high). By examining the model's performance at different temperature levels, we can gain insights into the trade-offs between linguistic diversity, grammatical accuracy, and the overall effectiveness of the model in correcting grammatical errors.

A low-temperature setting (e.g., 0.2) is expected to yield more conservative and grammatically correct outputs, as the model favors the most likely predictions based on its training data. However, this setting may limit the model's ability to generate diverse or creative corrections, potentially missing out on unconventional but valid alternatives [44].

On the other hand, a high temperature setting (e.g., 0.8) is likely to produce more varied and potentially innovative corrections, as the model explores a wider range of its output distribution. However, this increased diversity may come at the cost of decreased grammatical accuracy and coherence, as the model becomes more susceptible to generating nonsensical or ungrammatical outputs. The medium temperature setting (0.5) aims to strike a balance between diversity and grammatical correctness, potentially offering a reasonable trade-off between the extremes of low and high temperature settings.

By systematically evaluating the model's performance across these different temperature settings, we can gain valuable insights into the optimal configuration for the GEC task, considering factors such as grammatical accuracy, diversity of corrections, and overall task performance.

4. Experimental Settings

4.1. Dataset

We evaluate the K-GEC capability of ChatGPT using the K-NCT dataset [21]. The K-NCT dataset consists of both erroneous and correct sentences, spanning diverse domains,

phrases, and syllable counts, and encompasses varying counts and types of errors. Error sentences are annotated by marking the error locations, with each location specifically indicating the error type.

Table 2 provides the statistics of the K-NCT dataset. The dataset encompasses a total of 3000 sentences. On average, sentences containing errors are 43.27 characters in length, comprising an average word count of 10.39 and an average of 9.39 spaces. Conversely, correct sentences exhibit an average length of 43.29 characters, an average word count of 10.57, and an average of 9.57 spaces. These metrics highlight a notable similarity between erroneous and correct sentences, suggesting that the error-infused sentences in the K-NCT dataset adeptly emulate realistic mistakes while preserving the essence of the original sentence. We undertake a detailed analysis of the entire corpus provided by K-NCT, focusing on the 21 error types.

Table 2. Statistics of K-NCT dataset [21]. # of sents/tokens/words: number of sentences/tokens/words; △ avg of SL/WS/SS: average of sentence length/words/spaces per sentence.

K-NCT	Test	
	Error Sentence	**Correct Sentence**
# of sents	3000	3000
# of tokens	129,798	129,886
# of words	31,183	31,700
avg of SL △	43.27	43.29
avg of WS	10.39	10.57
avg of SS	9.39	9.57

4.2. Implementation Details

We analyze the performance of ChatGPT, powered by the gpt-3.5-turbo-0301 model, utilizing the OpenAI API. Experiments are conducted in both zero-shot and few-shots settings. In the few-shots configuration, we explore performance across four different numbers of examples: 1, 4, 8, and 16. We also investigate the influence of temperature on the model's performance by setting it to three distinct values: 0.2, 0.5, and 0.8. For performance assessment, we employ the BLEU [45] and GLEU [46] scores, widely used evaluation metrics in various spelling and grammar correction studies. BLEU is a metric used to assess the performance of machine translation systems by comparing the machine-translated output with human-translated references. It operates on the principle of n-grams (sequences of n consecutive words) and can be applied to any language pair. One advantage of BLEU is that it allows for rapid calculation. Similar to BLEU, there is another metric called GLEU, proposed by Napoles. GLEU requires human annotators to rewrite the original source sentence to create a corrected version. Unlike BLEU, GLEU takes the source sentence into account and is specifically designed to evaluate the performance of grammatical error correction systems.

5. Experimental Results

In this section, we delve into the performance of ChatGPT, focusing on two primary dimensions: explicit example information and temperature. Furthermore, through a qualitative analysis, we provide actual examples of K-GEC tasks executed by ChatGPT.

5.1. Main Results

Figure 1 delineates the mean performance of the model, considering variations in shots and temperature. Interestingly, the few-shots setup consistently outperforms the zero-shot approach. This observation underscores a pivotal insight: when LLMs like ChatGPT are tasked with K-GEC, supplementing the prompt with pertinent explicit examples can effectively harness and augment the inherent capabilities of the model, potentially leading to more accurate error corrections.

However, it is imperative to note that there is not a strict linear relationship between the number of shots and the positive influence they exert on performance. To illustrate, the performance differential between 1-shot and 4-shots is as pronounced as a maximum of 12.06 points on the BLEU metric. In contrast, the gap narrows significantly between 4-shots and 8-shots, with a difference of only up to 5.06 points. This indicates diminishing returns as the number of shots increases, suggesting an optimal threshold for shots that strikes a balance between computational efficiency and performance enhancement.

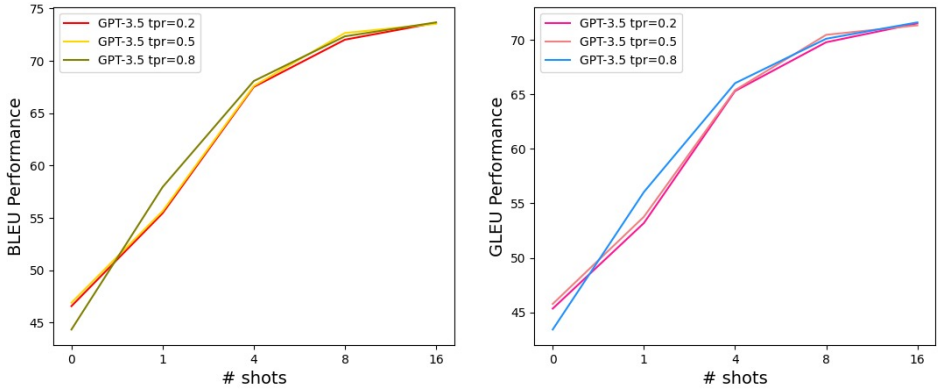

Figure 1. Performance of ChatGPT in K-GEC task with respect to temperature and shots. tpr indicates temperature.

Additionally, beyond the number of shots, the temperature setting stands as a pivotal parameter influencing ChatGPT's response generation. Intriguingly, performance differentials emerge even under identical shot settings when the temperature varies. For instance, in a zero-shot setting, a temperature of 0.5 yielded optimal results; however, when augmented with a single shot, the model's peak performance was observed at a temperature of 0.8. Moreover, as we increase the number of shots, the relative influence of temperature on performance seems to diminish. Such observations underscore the intricate interplay between explicit example information and temperature settings. It suggests that, by judiciously calibrating these parameters, one can effectively tap into and amplify the latent potential of large language models like ChatGPT, optimizing their capabilities for intricate tasks like K-GEC.

5.2. Additional Analysis

In Table 3, detailed performances across 21 error types are presented when the temperature is set to 0.2. Performances under zero-shot and four few-shots settings (1, 4, 8, 16) are elucidated. Due to a limited sample size in the dataset for the 'typing language' category, only zero-shot and 1-shot performances are considered.

In the zero-shot setting, the 'separation' category exhibited the highest performance, achieving a BLEU score of 64.90, which surpasses the average BLEU score of 46.58 by 18.32 points. Conversely, the 'typing language' category demonstrated the lowest performance with a BLEU score of 26.35, marking a 38.55-point difference from the highest score. Such disparities within the same settings indicate the necessity for a nuanced analysis to validate the capabilities of the LLM.

In all instances, it was observed that few-shot settings positively influenced the performance when compared to zero-shots. As the number of shots provided for error type correction increased, a corresponding enhancement in performance was noted. However, the magnitude of the influence varied across specific error types. For instance, the 'punctuation' category showed a significant performance increase with a difference of 41.11 points between 1-shot and 16-shots. While the performance in the 1-shot setting surpassed the average by BLEU 7.05 points, the 16-shot setting demonstrated a difference of 15.62 points, highlighting the efficacy of providing more shots. In contrast, the 'auxiliary predicate' category revealed a more muted effect, with a mere BLEU score difference of 12.90 between

1-shot and 16-shot settings. This difference is approximately 3.19 times less than that observed in the 'punctuation' category, suggesting that while increasing the number of shots generally has a positive effect on performance, the degree of influence is contingent upon the specific error type.

Table 3. K-GEC for ChatGPT-3.5 results for different error types. Avg. denotes the average of performance, and we set the temperature is 0.2.

Type	# Shots									
	0		1		4		8		16	
	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU
	temperature = 0.2									
spacing	47.59	47.62	60.17	58.31	76.97	74.37	79.40	76.99	79.68	77.36
punctuation	47.56	47.55	62.50	61.17	79.64	77.72	84.85	83.17	88.67	87.14
numerical	41.72	37.72	51.89	46.66	63.97	58.33	68.53	62.80	72.53	66.69
remove	42.37	41.27	53.75	50.90	60.02	57.54	66.86	63.85	69.63	67.03
addition	47.77	46.90	60.23	57.85	71.01	68.10	76.91	73.88	76.50	73.24
word_replace	46.29	45.10	52.67	50.77	71.14	69.58	67.68	65.43	70.08	68.07
rotation_replace	51.83	49.48	59.10	56.37	67.49	64.25	74.46	71.80	76.04	72.31
separation	64.90	66.27	71.49	70.85	86.41	85.49	87.06	86.54	85.39	85.24
typing_language	26.35	25.93	39.08	35.95	-	-	-	-	-	-
foreign_and_conversion	50.50	47.29	59.28	55.75	73.02	69.12	78.50	74.65	78.38	74.35
consonant_vowel_conversion	53.01	51.83	60.01	57.50	72.00	69.77	72.14	70.09	77.87	74.91
G2P	45.37	43.78	59.29	57.19	70.56	68.55	77.74	74.62	79.39	76.16
element	47.57	47.05	45.15	43.96	55.19	54.25	59.61	58.81	62.68	62.19
tense	37.60	31.56	50.08	43.49	62.75	53.46	65.32	56.12	67.15	58.92
postposition	53.93	50.19	62.74	59.25	68.45	65.45	78.32	75.73	77.11	73.97
suffix	49.07	48.99	58.67	57.92	74.03	74.15	75.57	75.64	75.61	76.57
auxiliary_predicate	51.44	51.80	56.74	57.78	64.63	64.32	72.43	74.76	69.64	70.49
dialect	49.54	53.02	52.82	55.92	62.39	67.86	67.55	72.26	69.84	74.09
polite_speech	43.43	42.92	54.03	53.81	58.18	58.87	64.86	65.15	67.69	69.12
behavior	43.66	38.95	43.58	36.83	52.79	47.71	57.35	51.46	59.80	56.01
neologism	36.62	37.00	51.18	48.78	59.57	57.24	65.27	62.13	69.02	66.46
Avg.	46.58	45.34	55.45	53.19	67.51	65.31	72.02	69.79	73.63	71.52

Tables 4 and 5 present the performance metrics under identical settings with temperatures set at 0.5 and 0.8, respectively. When assessed based on average performance, the influence of temperature appears to be relatively more subdued in comparison to the number of shots. However, nuances are observed across specific error types. For instance, replacement categories like 'word replace' and 'rotation replace' manifest enhanced performance as the temperature rises, maintaining a consistent shot setting (16 shots). Conversely, the 'tense' category evidences a decline in performance with an increase in temperature.

Table 4. K-GEC for ChatGPT-3.5 results for different error types. Avg. denotes the average of performance, and we set the temperature is 0.5.

Type	# Shots									
	0		1		4		8		16	
	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU
	temperature = 0.5									
spacing	45.93	45.84	63.32	61.40	76.31	74.07	78.89	76.18	80.22	77.71
punctuation	48.97	49.14	63.11	60.69	80.59	78.75	84.59	82.71	86.64	85.14
numerical	42.92	38.98	52.61	46.95	64.16	58.34	68.68	63.85	73.45	67.63

Table 4. *Cont.*

Type	# Shots									
	0		1		4		8		16	
	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU
remove	45.05	44.33	53.10	51.39	57.24	54.85	64.98	61.93	67.74	64.98
addition	48.01	47.37	59.14	56.99	68.10	65.59	76.28	73.18	76.31	73.33
word_replace	46.93	45.63	52.22	50.53	67.94	65.59	69.76	68.33	70.18	68.33
rotation_replace	51.43	50.93	58.72	55.92	67.19	64.21	77.97	74.20	76.61	73.66
separation	57.01	58.17	75.31	74.58	85.85	85.57	87.55	87.22	85.06	84.89
typing_language	44.16	39.80	27.12	29.72	-	-	-	-	-	-
foreign_and_conversion	47.68	45.16	58.23	56.03	75.20	71.59	77.18	72.88	80.34	76.43
element	38.59	38.85	47.83	46.45	56.84	56.39	63.99	63.11	64.13	63.52
consonant_vowel_conversion	52.55	50.96	60.50	57.91	70.98	68.88	73.87	71.69	78.15	75.53
G2P	47.40	45.44	59.93	57.17	71.93	69.98	75.36	73.12	77.64	74.77
tense	45.42	39.39	50.66	44.04	60.24	49.33	71.31	61.02	66.75	58.96
postposition	49.44	47.25	62.86	59.10	67.93	65.32	73.09	71.01	77.49	73.85
suffix	48.69	48.04	62.86	62.26	73.79	73.65	74.94	75.05	76.76	76.86
auxiliary_predicate	51.64	52.07	52.18	53.04	66.73	67.38	72.29	74.05	70.37	71.11
dialect	47.49	51.42	57.52	60.91	69.90	74.32	73.31	78.36	75.13	80.33
polite_speech	47.67	47.90	52.83	51.93	56.41	57.09	65.09	65.94	65.65	65.80
behavior	38.90	35.31	51.19	45.74	53.03	47.33	57.51	51.32	57.54	51.81
neologism	39.32	39.12	47.32	46.42	61.56	59.54	66.68	64.15	64.53	62.06
Avg.	46.91	45.77	55.65	53.77	67.60	65.39	72.66	70.47	73.53	71.34

Table 5. K-GEC for ChatGPT-3.5 results for different error types. Avg. denotes the average of performance, and we set the temperature is 0.8.

Type	# Shots									
	0		1		4		8		16	
	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU	BLEU	GLEU
temperature = 0.8										
spacing	44.92	44.73	63.36	61.68	76.93	74.69	79.84	77.39	80.13	77.50
punctuation	45.56	45.36	64.73	63.18	80.01	78.34	83.66	81.94	88.60	87.12
numerical	42.73	39.10	51.99	46.43	64.14	57.95	68.72	63.73	72.20	66.47
remove	43.34	42.50	52.79	50.71	57.63	55.56	65.14	63.06	67.59	64.70
addition	50.90	48.75	59.00	56.89	71.32	68.45	76.88	74.15	76.53	73.61
word_replace	45.94	44.90	51.01	49.61	67.32	65.18	70.07	67.99	71.96	70.35
rotation_replace	46.89	45.79	62.89	59.35	70.81	66.73	76.71	73.22	77.82	74.64
separation	59.41	61.52	77.39	76.83	84.46	84.31	86.49	86.31	86.33	85.78
typing_language	20.93	20.93	61.87	61.22	-	-	-	-	-	-
foreign_and_conversion	45.36	43.96	59.78	57.35	74.87	71.93	77.86	74.21	80.34	76.86
consonant_vowel_conversion	52.20	50.91	60.84	57.45	72.11	70.07	73.66	71.75	77.74	75.16
G2P	46.70	44.75	58.45	55.62	71.74	70.47	79.01	76.23	79.11	76.27
element	39.82	40.17	46.21	46.43	54.56	53.89	58.73	57.24	58.17	57.43
tense	44.35	37.55	50.76	44.61	61.49	51.29	65.87	54.93	68.02	59.35
postposition	41.25	39.24	65.25	61.75	72.38	70.61	76.11	72.90	75.71	73.52
suffix	45.92	45.43	64.13	63.58	74.10	74.97	75.66	75.58	80.13	80.61
auxiliary_predicate	48.12	50.23	54.65	56.04	66.50	66.98	72.46	74.45	75.15	76.73
dialect	44.77	48.64	60.04	63.95	69.46	74.49	72.41	77.33	71.20	76.07
polite_speech	44.85	43.91	52.59	51.89	56.71	56.63	61.61	62.10	66.05	66.81
behavior	39.52	36.11	47.01	42.80	55.91	51.35	59.80	55.11	54.89	49.94
neologism	37.92	37.40	52.26	49.51	58.96	56.74	66.04	62.67	65.61	63.21
Avg.	44.35	43.42	57.95	56.04	68.07	66.03	72.34	70.11	73.67	71.61

For more subtle errors, more complex outcomes are observed. For instance, in the case of polite speech errors, as the number of shots increases, the influence of temperature diminishes. In zero-shot scenarios, there is a performance variation of up to 4.24 compared to temperatures set at 0.2, whereas in 16-shot scenarios, the difference is at most 2.04. Additionally, concerning behavior errors, providing shots proves effective; however, it is observed that temperature tends to decrease performance as the number of shots increases. This underscores the necessity for a nuanced evaluation, segmenting by specific error types, to acquire a more comprehensive understanding of the capabilities of LLMs, including ChatGPT.

5.3. Quantitative Analysis

In Table 6, we present illustrative examples of K-GEC tasks performed by ChatGPT. The 'source' column illustrates sentences containing errors, while the 'target' column provides the corresponding corrected versions. The table further categorizes the results based on varying 'temperature' and 'shots' parameters. One noteworthy category of error is the 'punctuation' type, where the performance variation is particularly pronounced across different shots. In the exemplified case, the inadvertent insertion of exclamation marks in the midst of sentences results in an error. With a consistent temperature of 0.2, the performance exhibits significant variations based on the number of shots. In the zero-shot instance, ChatGPT successfully removes the misplaced exclamation marks but overcorrects by introducing superfluous terms such as '쓰인 (written)'. Similar overcorrections are observed in both 1-shot and 4-shot scenarios, with additions like '적힌 (written)' and '쓴 (wrote it)'. However, the 8-shot scenario demonstrates a seamless correction aligning precisely with the target sentence. This evidence underscores the considerable performance variability of ChatGPT depending on the number of shots. In particular, it is noteworthy that such results emerged even when the temperature was set low, which is generally expected to produce more grammatically correct and deterministic text.

Table 6. Qualitative analysis results. Source refers to the original or input text that contains spelling errors or mistakes. Target represents the desired or corrected version of the text. ChatGPT shows the results of performing K-GEC based on temperature and shots.

	Sentence	Temperature	# Shots
	Error type = punctuation		
Source	그녀는 봉투의 보내는! 사람을 보고 누가 썼는지 알 수 있습니다. (She can identify who wrote it by looking at the sender! of the envelope.)	-	-
Target	그녀는 봉투의 보내는 사람을 보고 누가 썼는지 알 수 있습니다. (She can identify who wrote it by looking at the sender of the envelope.)	-	-
ChatGPT	그녀는 봉투에 쓰인 보내는 사람을 보고 누가 썼는지 알 수 있습니다. (She can identify who wrote it by looking at the sender written on the envelope.)	0.2	0
	그녀는 봉투에 적힌 보내는 사람을 보고 누가 썼는지를 알 수 있습니다. (She can identify who wrote it by looking at the sender written on the envelope.)	0.2	1
	그녀는 봉투에 쓴 사람을 보고 누가 보냈는지 알 수 있습니다. (She can identify who sent it by looking at the person who wrote it on the envelope.)	0.2	4
	그녀는 봉투의 보내는 사람을 보고 누가 썼는지 알 수 있습니다 (She can identify who wrote it by looking at the sender of the envelope)	0.2	8
	그녀는 봉투의 보내는 사람을 보고 누가 썼는지 알 수 있습니다. (She can identify who wrote it by looking at the sender of the envelope.)	0.2	16
	Error type = tense		
Source	내일 몇 시쯤 끝났을 것 같아요? (What time do you think it was over tomorrow?)	-	-
Target	내일 몇 시쯤 끝날 것 같아요? (What time do you think it will be over tomorrow?)	-	-
ChatGPT	내일 몇 시쯤 끝날 것 같아요? (What time do you think it will be over tomorrow?)	0.2	1
	내일 몇 시쯤 끝날 것 같아요? (What time do you think it will be over tomorrow?)	0.5	1
	내일 몇 시쯤 끝나는 걸까요? (I wonder what time it will be finished tomorrow.)	0.8	1

Table 6. *Cont.*

	Sentence	Temperature	# Shots
	Error type = neologism		
Source	너는 머 타고 집에 갈 거니? (W will you ride to go home?)	-	-
Target	너는 뭐 타고 집에 갈 거니? (What will you ride to go home?)	-	-
Commercialization System	교정된 내용이 없습니다. (No corrections are needed.)	-	-
ChatGPT	너는 뭐 타고 집에 갈 거니? (What will you ride to go home?)	0.2	8
	Error type = remove		
Source	저는 당신과 당장 결호했으면 합니다. (I would like to mary you immediately.)	-	-
Target	저는 당신과 당장 결혼했으면 합니다. (I would like to marry you immediately.)	-	-
Commercialization System	저는 당신과 당장 결혼했으면 합니다. (I would like to marry you immediately.)	-	-
ChatGPT	저는 당신과 당장 결혼하고 싶습니다. (I desire to marry you immediately.)	0.2	8

For the 'tense' type of errors, there's a notable difference in performance depending on the temperature setting, even when the number of shots remains consistent. In the given examples, the K-GEC outcomes based on varying temperatures are demonstrated under the same 1-shot setting. With temperatures set at 0.2 and 0.5, the corrections align closely with the target sentence. However, a setting of 0.8 for the temperature appears to introduce changes in the sentence structure. In grammar correction tasks, the source and target sentences are typically very similar, with errors manifesting in only some parts of the sentence. Hence, it is vital to avoid overcorrection. In this context, the linguistic fluency of ChatGPT sometimes emerges as a drawback. Nonetheless, there are instances where understanding the context and modifying word expressions accordingly are essential, suggesting that linguistic fluency is not always unnecessary. Therefore, it is crucial to discern the capabilities and types required for the task at hand and to validate these against the abilities of the LLM.

The examples provided for 'neologism' and 'remove' types serve as a comparison between the correction results of a commercialized system and those produced by ChatGPT. The commercialized system was chosen based on its superior performance in Korean grammar correction https://bit.ly/3CtaPl4. The ChatGPT results were derived with a temperature setting of 0.2 and 8 shots.

For the 'neologism' type, the commercialized system failed to detect and consequently correct the error. In contrast, ChatGPT successfully identified and rectified the neologism error. This capability can be attributed to the provision of explicit examples and descriptions regarding the neologism type of ChatGPT. This suggests that ChatGPT can display commendable performance with new types of errors using just instructions, without the need for additional training. However, for the 'remove' type, while the commercialized system effectively corrected the text in alignment with the target sentence, ChatGPT overcorrected. This can be interpreted similarly to the 'tense' type, where linguistic fluency might have inadvertently introduced negative consequences. Thus, it is evident that ChatGPT's potential abilities are influenced by explicit example information and temperature settings. This demonstrates that adjusting the few-shots or temperature is an effective method to enhance ChatGPT's efficacy in K-GEC tasks. Moreover, given that this influence varies based on the error type, a nuanced analysis is imperative.

6. Conclusions

Our analysis of ChatGPT's performance on the Korean Grammatical Error Correction (K-GEC) task revealed both the strengths and inherent challenges when dealing with the intricate nuances of the Korean language. We demonstrate the effectiveness of strategies to enhance ChatGPT's performance on K-GEC tasks. For instance, modifications in parame-

ters, particularly focusing on shots and temperature adjustments, significantly impacted the model's outputs. As the NLP domain advances with cutting-edge language models like ChatGPT, it is crucial to recognize and understand both their potential and their boundaries. This study provides insights into the capabilities of current large language models (LLMs) for linguistically rich tasks, emphasizing the continual need for research and improvements in the field.

Grammatical error correction (GEC) systems are vital for various practical applications and services. In the realm of education, GEC can assist students and language learners in improving their writing skills by providing accurate and contextual feedback on grammatical errors. Businesses and organizations can benefit from GEC tools to ensure the quality and professionalism of their written communications, such as reports, marketing materials, and customer interactions.

Moreover, as digital assistants and conversational AI systems become more prevalent, integrating robust GEC capabilities is essential to enhance the user experience and maintain the quality of a generated text. Language models like ChatGPT have the potential to power such assistants, but their performance on complex linguistic tasks, like GEC for morphologically rich languages like Korean, must be thoroughly evaluated and improved. By understanding the strengths and limitations of current LLMs in handling intricate grammatical structures and nuances, researchers and developers can work towards refining these models and developing more sophisticated techniques for GEC. This includes exploring techniques such as transfer learning, data augmentation, and incorporating linguistic knowledge into the models.

Furthermore, the development of LLM-based GEC systems for various languages can contribute to bridging language barriers and promoting linguistic diversity in the digital realm. As technology continues to shape our communication landscape, ensuring accurate and grammatically correct text across languages becomes increasingly crucial for fostering understanding and inclusivity.

In summary, this study not only sheds light on the performance of ChatGPT on the K-GEC task but also highlights the broader significance of LLM-based GEC systems for practical applications and services. By continuously advancing research in this field, we can unlock the full potential of language models, enabling more accurate and contextual grammatical error correction capabilities across various domains and languages.

Author Contributions: Conceptualization, C.P. and S.K.; methodology, C.P.; software, C.P., S.K. and G.K.; validation, C.P.; formal analysis, S.K.; investigation, C.P. and S.K.; resources, C.P. and G.K.; data curation, S.K.; writing—original draft preparation/review and editing, C.P., S.K. and G.K.; visualization, S.K. and G.K.; supervision/project administration/funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: A publicly available dataset was utilized in this study. These data can be found here: "https://github.com/seonminkoo/K-NCT" (accessed on1 October 2023).

Conflicts of Interest: Author Chanjun Park was employed by the company Upstage. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Park, C.; Kim, K.; Yang, Y.; Kang, M.; Lim, H. Neural spelling correction: Translating incorrect sentences to correct sentences for multimedia. *Multimed. Tools Appl.* **2020**, *80*, 34591–34608. [CrossRef]
2. Wang, Y.; Wang, Y.; Liu, J.; Liu, Z. A comprehensive survey of grammar error correction. *arXiv* **2020**, arXiv:2005.06600.

3. Lee, J.H.; Kwon, H.C. Context-Sensitive Spelling Error Correction Techniques in Korean Documents using Generative Adversarial Network. *J. Korea Multimed. Soc.* **2021**, *24*, 1391–1402.
4. Xiong, J.; Zhang, Q.; Zhang, S.; Hou, J.; Cheng, X. HANSpeller: A unified framework for Chinese spelling correction. *Int. J. Comput. Linguist. Chin. Lang. Process.* **2015**, *20*, 1.
5. Kim, M.; Jin, J.; Kwon, H.C.; Yoon, A. Statistical context-sensitive spelling correction using typing error rate. In Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering, Sydney, Australia, 3–5 December 2013; pp. 1242–1246.
6. Lee, J.H.; Kim, M.; Kwon, H.C. Improved statistical language model for context-sensitive spelling error candidates. *J. Korea Multimed. Soc.* **2017**, *20*, 371–381. [CrossRef]
7. Lee, M.; Shin, H.; Lee, D.; Choi, S.P. Korean Grammatical Error Correction Based on Transformer with Copying Mechanisms and Grammatical Noise Implantation Methods. *Sensors* **2021**, *21*, 2658. [CrossRef]
8. Park, C.; Park, S.; Lim, H. Self-Supervised Korean Spelling Correction via Denoising Transformer. In Proceedings of the 2023 International Conference on Information, System and Convergence Applications 2020.
9. Park, C.; Seo, J.; Lee, S.; Son, J.; Moon, H.; Eo, S.; Lee, C.; Lim, H.S. Hyper-BTS Dataset: Scalability and Enhanced Analysis of Back TranScription (BTS) for ASR Post-Processing. In Proceedings of the Findings of the Association for Computational Linguistics: EACL 2024, St. Julian's, Malta, 18–22 March 2024; pp. 67–78.
10. OpenAI-Blog. ChatGPT: Optimizing Language Models for Dialogue. 2022. Available online: https://chatgpt.r4wand.eu.org/ (accessed on1 November 2023).
11. Zhao, W.X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. A Survey of Large Language Models. *arXiv* **2023**, arXiv:2303.18223.
12. Kim, D.; Park, C.; Kim, S.; Lee, W.; Song, W.; Kim, Y.; Kim, H.; Kim, Y.; Lee, H.; Kim, J.; et al. Solar 10.7 b: Scaling large language models with simple yet effective depth up-scaling. *arXiv* **2023**, arXiv:2312.15166.
13. Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. Gpt-4 technical report. *arXiv* **2023**, arXiv:2303.08774.
14. Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A.M.; Hauth, A.; et al. Gemini: A family of highly capable multimodal models. *arXiv* **2023**, arXiv:2312.11805.
15. Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.d.l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825.
16. OpenAI. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
17. Liang, Y.; Wu, C.; Song, T.; Wu, W.; Xia, Y.; Liu, Y.; Ou, Y.; Lu, S.; Ji, L.; Mao, S.; et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *arXiv* **2023**, arXiv:2303.16434.
18. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
19. Rozovskaya, A.; Roth, D. Grammar error correction in morphologically rich languages: The case of Russian. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 1–17. [CrossRef]
20. Imamura, K.; Saito, K.; Sadamitsu, K.; Nishikawa, H. Grammar error correction using pseudo-error sentences and domain adaptation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Korea, 8–14 July 2012; Volume 2, pp. 388–392.
21. Koo, S.; Park, C.; Seo, J.; Lee, S.; Moon, H.; Lee, J.; Lim, H. K-nct: Korean neural grammatical error correction gold-standard test set using novel error type classification criteria. *IEEE Access* **2022**, *10*, 118167–118175. [CrossRef]
22. Koo, S.; Park, C.; Kim, J.; Seo, J.; Eo, S.; Moon, H.; Lim, H.S. KEBAP: Korean Error Explainable Benchmark Dataset for ASR and Post-processing. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 4798–4815.
23. Koo, S.; Park, C.; Kim, J.; Seo, J.; Eo, S.; Moon, H.; Lim, H. Toward Practical Automatic Speech Recognition and Post-Processing: A Call for Explainable Error Benchmark Guideline. *arXiv* **2024**, arXiv:2401.14625.
24. Li, H.; Wang, Y.; Liu, X.; Sheng, Z.; Wei, S. Spelling error correction using a nested rnn model and pseudo training data. *arXiv* **2018**, arXiv:1811.00238.
25. Solyman, A.; Wang, Z.; Tao, Q. Proposed model for arabic grammar error correction based on convolutional neural network. In Proceedings of the 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Khartoum, Sudan, 21–23 September 2019; pp. 1–6.
26. Kuznetsov, A.; Urdiales, H. Spelling Correction with Denoising Transformer. *arXiv* **2021**, arXiv:2105.05977.
27. Tarnavskyi, M.; Chernodub, A.; Omelianchuk, K. 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, Dublin, Ireland, 22–27 May 2022; Volume 1: Long Papers, pp. 3842–3852. [CrossRef]
28. Kaneko, M.; Takase, S.; Niwa, A.; Okazaki, N. Interpretability for Language Learners Using Example-Based Grammatical Error Correction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, 22–27 May 2022; pp. 7176–7187. [CrossRef]

29. Gan, Z.; Xu, H.; Zan, H. Self-Supervised Curriculum Learning for Spelling Error Correction. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 3487–3494.
30. Cao, H.; Yang, W.; Ng, H.T. Grammatical Error Correction with Contrastive Learning in Low Error Density Domains. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 4867–4874. [CrossRef]
31. Sun, X.; Ge, T.; Wei, F.; Wang, H. Instantaneous grammatical error correction with shallow aggressive decoding. *arXiv* **2021**, arXiv:2106.04970.
32. Wang, D.; Song, Y.; Li, J.; Han, J.; Zhang, H. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2517–2527. [CrossRef]
33. Gudmundsson, J.; Menkes, F. Swedish Natural Language Processing with Long Short-Term Memory Neural Networks: A Machine Learning-powered Grammar and Spell-Checker for the Swedish Language. Bachelor's Thesis, Linnaeus University, Växjö, Sweden, 2018.
34. Náplava, J.; Popel, M.; Straka, M.; Straková, J. Understanding Model Robustness to User-generated Noisy Texts. In Proceedings of the Seventh Workshop on Noisy User-Generated Text (W-NUT 2021), Online, 11 November 2021; pp. 340–350. [CrossRef]
35. Hidayatullah, E. Evaluating the effectiveness of ChatGPT to improve English students' writing skills. *Humanit. Educ. Appl. Linguist. Lang. Teaching Conf. Ser.* **2024**, *1*, 14–19.
36. Schmidt-Fajlik, R. Chatgpt as a grammar checker for japanese english language learners: A comparison with grammarly and prowritingaid. *AsiaCALL Online J.* **2023**, *14*, 105–119. [CrossRef]
37. Li, Y.; Huang, H.; Ma, S.; Jiang, Y.; Li, Y.; Zhou, F.; Zheng, H.T.; Zhou, Q. On the (in) effectiveness of large language models for chinese text correction. *arXiv* **2023**, arXiv:2307.09007.
38. Zhang, J.; Feng, H.; Liu, B.; Zhao, D. Survey of Technology in Network Security Situation Awareness. *Sensors* **2023**, *23*, 2608. [CrossRef] [PubMed]
39. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
40. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
41. Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T.B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; Amodei, D. Scaling laws for neural language models. *arXiv* **2020**, arXiv:2001.08361.
42. Wei, X.; Cui, X.; Cheng, N.; Wang, X.; Zhang, X.; Huang, S.; Xie, P.; Xu, J.; Chen, Y.; Zhang, M.; et al. Zero-shot information extraction via chatting with chatgpt. *arXiv* **2023**, arXiv:2302.10205.
43. Peng, K.; Ding, L.; Zhong, Q.; Shen, L.; Liu, X.; Zhang, M.; Ouyang, Y.; Tao, D. Towards making the most of chatgpt for machine translation. *arXiv* **2023**, arXiv:2303.13780.
44. Ippolito, D.; Kriz, R.; Sedoc, J.; Kustikova, M.; Callison-Burch, C. Comparison of Diverse Decoding Methods from Conditional Language Models. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3752–3762. [CrossRef]
45. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 6–12 July 2002; pp. 311–318.
46. Napoles, C.; Sakaguchi, K.; Post, M.; Tetreault, J. Ground truth for grammatical error correction metrics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 2, pp. 588–593.