ETA: Enriching Typos Automatically from Real-World Corpora for Few-Shot Learning

Anonymous ACL submission

Abstract

Spell checking is the task of rectifying errors in a sentence resulting from various factors, and despite continuous research in this field, research often focused on widely known specific languages. In this study, we focus on the Korean language and its linguistic characteristics, particularly the propensity for a single character can be incorrect in diverse ways. Therefore, we categorize spelling errors from real-world corpora and automatically construct an error corpus based on their statistical patterns. When we employed them to leverage the impact of a pre-trained large language model (LLM), we confirm that utilizing the introduced spelling errors as samples for few-shot learning can be helpful in error correction tasks. We hope that this study contributes to the automatic construction of error corpora and prompt-based approaches for other low-resource languages.

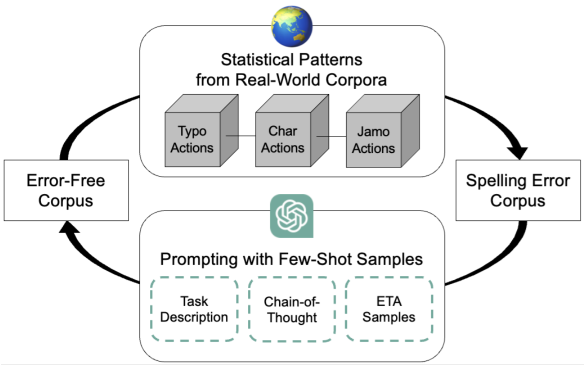

Figure 1: Process that automatically enriches spelling errors with their statistical patterns to construct an error corpus, and then corrects them through prompting using few-shot samples of those spelling errors.

1 Introduction

Spell checking serves as the process of correcting spelling errors within a given sentence and can be used as a post-processing task in various natural language processing applications to ensure sentence clarity (Liao et al., 2023; Pan et al., 2022; Kwon et al., 2021). This necessity extends beyond widely known languages, such as English, inspiring interest in low-resource languages and their specific research (Abdulrahman and Hassani, 2022; Wiechetek et al., 2021). To delve into spell checking for low-resource languages, it is imperative to conduct a comprehensive examination of the linguistic characteristics inherent to each language.

While Korean has experienced a year-over-year increase in global usage (Lusin et al., 2023), its linguistic features remain unexplored in spell checking task. We note that the unique writing system in Korean allows a wide range of typos, even within a single character. Each character in Korean adheres to the C1VC2 form (Song, 2006), where C1 represents the initial sound, V represents the middle sound, and C2 represents the optional final sound. For example, the character '녕' from the word '안녕하세요(Hello)' is composed of 'ㄴ', 'ㅕ', and 'ㅇ'. Theoretically, there can be 19, 21, and 28 candidates for each of these components (Lee, 2006), yielding a total of 11,172 possible combinations within a single character.

Owing to these possibilities, it is inefficient to consider all kinds of spelling errors, so we hypothesize that people make certain kinds of errors more frequently. Therefore, we categorize spelling errors from real-world corpora collected online, referred to as *Typo Actions*, and leverage their statistical patterns to construct a corpus with spelling errors. While existing studies in Korean have used grammatical errors from language learners (Yoon et al., 2023), deliberately introduced noises through textual variants (Lee et al., 2021; Min et al., 2020), or parallel datasets created by human annotators (Koo et al., 2022), none of them have integrated spelling errors with statistical patterns comparable to our work. The spelling errors we introduce are automatically incorporated in the form of typos, without requiring the need for human annotators.

We evaluate the effectiveness of utilizing these spelling errors by prompting them to a large language model (LLM). Prompt-based methods for few-shot learning have been proposed to exploit the capabilities of LLMs (Zhao et al., 2023; Brown et al., 2020), and current studies have also utilized prompting in error correction tasks (Loem et al., 2023; Fang et al., 2023; Khondaker et al., 2023), but there has been a lack of analysis considering the potential scaling for spelling errors in Korean. Therefore, we examine the changes in the use of spelling errors within the LLM by including a task description, zero-shot chain-of-thought (CoT) (Kojima et al., 2022), and sentence pairs featuring our spelling errors as few-shot samples in the prompt. The overall process we are introducing is illustrated in Figure 1. We briefly summarize the contributions of this work.

- We introduce the statistical patterns of spelling errors from real-world corpora, called *Typo Actions*, and employ them for the automatic construction of a parallel corpus. This provides a pragmatic and sensible way to generate typos within a low-resource language.

- We experiment few-shot learning with CoT by incorporating spelling errors while prompting the LLM, and as a result, we suggest that our spelling errors can be helpful to the LLM for spell checking task.

- By adjusting the inclusion rate of spelling errors in the process of leveraging the introduced process, we conduct various analyses of their results with few-shot learning.

Algorithm 1 Enriching Typos Automatically

```
for error-free word in Error-Free Sentence do
    Insert error-free word to candidates

    while C > 0 do
        Act ← one of Typo Actions
        if Act == Character Actions then
            Act ← one of Character Actions
        else
            Act ← one of Jamo Actions
        error ← error-free word + Act
        Insert error to candidates
        C ← C − 1

for each of the candidates per error-free word do
    if prob ~ U[0, 1] < P_ensure then
        word' ← error-free word
    else
        word' ← one of errors
    Insert word' to Error Sentence

Repeat for all Error-Free Sentences
```

2 Method

2.1 Typo Actions

We introduce the types of spelling errors referred to as *Typo Actions*, which are categorized into *Character/Jamo Actions*[1]. Further details containing the referenced real-world corpora and distributions of *Typo Actions* are provided in Appendix A.

The former comprises two components: the absence or addition of a specific character. These are denoted as *add_char* and *del_char*, respectively, as both cases involve either the addition or deletion of a specific character to correct the sentence. The latter comprises six components: incorrect sounds for each or any of the initial, middle, and final sounds. These are denoted using *C1*, *V*, and *C2*, depending on which sound is incorrect. For example, if only the initial sound is incorrect, it is referred to as *C1*, and if both the middle and final sounds are incorrect, it is referred to as *V+C2*.

We devise the process of introducing errors into a sentence using their statistical patterns, as described in Algorithm 1. Initially, word tokenization is conducted to determine whether to generate an error for each word. Rather than simply resulting in just one error per word, we define a capacity C to allow multiple distinct errors into candidates. To prevent an excessive number of errors, an error-free word is also included in the candidates. Consequently, if an error-free sentence consists of n words, we generate n sets of candidates, resulting in $n \times (C+1)$ error-free words and errors.

To construct an error sentence, we have the option to select either an error-free word or errors from each of the candidates per word. In this scenario, to reduce the likelihood of a high error rate in a sentence, we define a probability P_{ensure} to guarantee the selection of an error-free word. Consequently, the probability of selecting each error is $(1 - P_{ensure})/C$. This procedure is repeated for all error-free sentences, resulting in an error corpus that incorporates real-world statistical patterns[2].

2.2 Prompt Design

We devise various prompt designs, including samples that incorporate the introduced spelling errors, to conduct spell checking with the LLM. Especially

[1]Both consonants and vowels are referred to as *jamo* in Korean, which are denoted as *C1*, *V*, and *C2* in this study.

[2]We set C to 3 and P_{ensure} to 0.6.

Method	Word			Character			Average		
	P	R	F1	P	R	F1	P	R	F1
task description	61.90	61.91	61.86	**62.00**	**61.94**	**61.93**	61.95	61.93	61.89
task description + CoT	60.97	61.10	60.97	60.65	60.79	60.66	60.81	60.94	60.81
task description + CoT + ETA 1-shot	62.35	62.39	62.31	61.33	61.41	61.31	61.84	61.89	61.81
task description + CoT + ETA 4-shot	62.56	62.56	62.50	60.72	60.78	60.70	61.64	61.67	61.60
task description + CoT + ETA 8-shot	**62.97**	**62.90**	**62.87**	60.98	60.98	60.93	**61.98**	**61.94**	**61.91**

Table 1: Experimental results of correction the error corpus into an error-free corpus using the introduced spelling errors. P, R, and F1 represent precision, recall, f1-score, respectively. Average presents the combined result for word and character metrics. When spelling errors were incorporated into both the test set and the few-shot samples, the P_{ensure} was set to 0.6.

Method	Word			Character			Average		
	P	R	F1	P	R	F1	P	R	F1
task description + CoT + ETA 1-shot	62.45	62.48	62.41	**61.35**	**61.42**	**61.33**	61.90	**61.95**	**61.87**
task description + CoT + ETA 4-shot	62.61	62.60	62.55	60.63	60.68	60.60	61.62	61.64	61.57
task description + CoT + ETA 8-shot	**62.92**	**62.86**	**62.83**	60.96	60.96	60.91	**61.94**	61.91	**61.87**

Table 2: Experimental results of correction the error corpus into an error-free corpus using the introduced spelling errors. When spelling errors were incorporated for the experiment, the test set had a P_{ensure} of 0.6 and the few-shot samples had a P_{ensure} of 0.3.

in spell checking and grammatical error correction tasks, the problem of over-correction arises, which is the unnecessary modification of the correct words in a given sentence instead of correcting errors (Wu et al., 2023; Al-Sabahi and Yang, 2023). Therefore, we write $\texttt{text}_{task}$ based on this for the task description.

We take inspiration from the zero-shot CoT (Kojima et al., 2022), so we incorporated some texts to enhance reasoning for spell checking. This $\texttt{text}_{cot}$ is presented after the task description. The two above prompts are defined as follows, and the input error sentence is placed in the `input`.

$$P_{task} = \text{`}\texttt{text}_{task}; \texttt{input: output:}\text{'}, \quad (1)$$

$$P_{cot} = \text{`}\texttt{text}_{task}; \texttt{text}_{cot}; \texttt{input: output:}\text{'}, \quad (2)$$

Following this, we engage in few-shot learning (Brown et al., 2020) using samples that contain the introduced spelling errors. The $\texttt{text}_{n\text{-}shot}$, stating that samples are available for inference, and `samples` that forms of the n samples are contained. The prompt is defined as follows, and the n-shot samples and the input error sentence are placed in the `samples` and `input`, respectively.

$$P_{n\text{-}shot} = \text{`}\texttt{text}_{task}; \texttt{text}_{cot}; \texttt{text}_{n\text{-}shot}; \texttt{samples}; \texttt{input: output:}\text{'}. \quad (3)$$

The actual texts employed in all prompts and the specific procedure of selecting samples for few-shot learning are detailed in Appendix B.

3 Experiments

3.1 Dataset

We collected 500k sentences from the Korean Wikipedia[3] and constructed an error corpus using the proposed process. We split the dataset into train, validation, and test sets in the ratio of 8:1:1. We used the train and validation sets to select few-shot learning samples.

3.2 Experimental Results

We present the results of the prompts utilized for correcting the spelling errors in an error-free form in Table 1. The best performances for each metric and averages across word and character distinctions are highlighted in bold.

Examining the word-level results, we observed that few-shot learning with spelling errors as samples leads to a modest performance enhancement for all metrics. It was likely attributed to the introduction of spelling errors based on word tokenization during the construction of the error corpus. As more relevant samples were incorporated, a slight increase in performance was also observed.

However, when considering the character-level results, we confirmed that there were marginal improvements when utilizing only task descriptions. The Wikipedia texts we used are more susceptible to spelling errors, primarily owing to the diverse proper nouns. Consequently, the scenario in which we prioritized task descriptions over provid-

[3]https://dumps.wikimedia.org/kowiki/

IQRs	*EF* vs. 0.6	*EF* vs. 0.3	0.6 vs. 0.3
75%	84.73	84.64	79.17
50%	75.03	74.98	71.60
25%	62.89	62.92	62.48

Table 3: IQR ranges of sentence similarities between error-free and spelling error sentences. *EF* stands for the error-free corpus, with each float value representing a spelling error corpus constructed according to P_{ensure}.

ing additional texts yielded better results during the character-level evaluation.

In the context of spell checking, it is crucial to consider not only the word or character level individually but both of them. Therefore, when we average the results, finally we observed that few-shot learning with the introduced spelling errors outperformed other prompts on all metrics.

3.3 Adjusting Difficulty

We compared the results when more challenging samples were presented with few-shot learning, so we adjusted the P_{ensure}, which was used to introduce spelling errors. Thus, by setting P_{ensure} to a lower value, we additionally constructed an error corpus that reduced the likelihood of selecting error-free words[4].

We present the results that maintain the same test set as Table 1 but modified the samples for few-shot learning to be more challenging in Table 2. The results at both the word and character-level exhibited similar trends to the previous experiments. However, in terms of the variation in performance, slightly improved results were observed when the `samples` and `input` contained similar degrees of spelling errors.

We assumed that sentences are more challenging as they become noisier due to the formation of spelling errors. Therefore, we compared the similarity of sentences across situations and represented the distribution through interquartile (IQR) ranges, as shown in Table 3. We employed a Sentence-BERT (Reimers and Gurevych, 2019) pre-trained on Korean texts[5] to obtain sentence embeddings. When comparing the existence of spelling errors, we observed that sentence similarity decreases slightly when P_{ensure} was reduced from 0.6 to 0.3. Therefore, selecting error-free words with a lower probability led to a more divergent from the error-free sentence. Additionally, when comparing only the sentences with spelling errors, we discovered that a P_{ensure} of 0.3 retained only 71% of the meaning compared to a sentence with a P_{ensure} of 0.6. Consequently, the introduced spelling errors could impact to recognition of sentence meaning, and this aspect would be inherent in the correction process.

[4] We set P_{ensure} to 0.3.

[5] https://github.com/snunlp/KR-SBERT

4 Related Work

Comprehending the intent or context of a sentence is crucial for spell checking (Anderson-Inman and Knox-Quinn, 1996; Mitton, 1987). Text matching methods such as n-gram analysis or dictionary lookup have been conducted (Randhawa and Saroa, 2014). However, these rule-based methods have limitations in addressing the meaning of the sentence, so RNN, BERT, and other transformer-based models have been proposed to detect and correct errors in a sentence (Zhu et al., 2022; Ji et al., 2021; Zhang et al., 2020; Zaky and Romadhony, 2019; Etoori et al., 2018).

For the Korean language, error corpora have been created by introducing noise manually and adopting the above model structures (Lee et al., 2021; Min et al., 2020). Human annotators have been employed to introduce spelling and grammar errors (Koo et al., 2022), or datasets have been proposed from language learner corpora to categorize various error types (Yoon et al., 2023).

More recently, as pre-trained LLMs have been proposed, studies have examined the effects of prompts on the performance of tasks. Researchers have incorporated CoT into zero-shot learning and conducted comparative analysis for samples with few-shot learning (Loem et al., 2023; Fang et al., 2023). There have also been investigations extending few-shot learning to low-resource languages (Khondaker et al., 2023; Elsner and Needle, 2023; Schneider et al., 2022).

5 Conclusion

We propose a method for utilizing spelling errors present in real-world corpora and constructing an error corpus based on automated process by statistical patterns of them. When we conducted experiments to assess their impact on few-shot learning, we confirmed that it can be helpful for error correction task when prompted with samples that contain the introduced spelling errors. We further plan to explore methods for validating spelling errors and designing tailored prompts to use them.

Limitations

Our procedure to construct an error corpus cannot be directly applied to other languages since it generates typos according to the unique writing system in Korean. However, by referring to this automation process that uses linguistic features, we believe that other low-resource researchers can develop their own corpora. We should rely on the specific real-world corpora to reflect spelling errors. From this point of view, we expect that more online texts will be collected for extensive utilization.

Ethics Statement

We generate and employ spelling errors based on their online occurrences, emphasizing that their distribution originates from authentic online sources. Additionally, despite the active use of prompting with few-shot samples, employing a pre-trained LLM might introduce inherent bias in the model output. This should be considered when developing our research or expanding it to other languages.

References

Roshna Abdulrahman and Hossein Hassani. 2022. A language model for spell checking of educational texts in kurdish (sorani). In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 189–198.

Kamal Al-Sabahi and Kang Yang. 2023. Supervised copy mechanism for grammatical error correction. *IEEE Access*, 11:72374–72383.

Lynne Anderson-Inman and Carolyn Knox-Quinn. 1996. Spell checking strategies for successful students. *Journal of Adolescent Adult Literacy*, 39(6):500–503.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Micha Elsner and Jordan Needle. 2023. Translating a low-resource language using gpt-3 and a human-readable dictionary. In *Proceedings of the 20th SIGMORPHON workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 1–13.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic spelling correction for resource-scarce languages using deep learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152, Melbourne, Australia. Association for Computational Linguistics.

Tao Fang, Shu Yang, Kaixin Lan, Derek F Wong, Jinpeng Hu, Lidia S Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation. *arXiv preprint arXiv:2304.01746*.

Tuo Ji, Hang Yan, and Xipeng Qiu. 2021. SpellBERT: A lightweight pretrained model for Chinese spelling check. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3544–3551, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Md Tawkat Islam Khondaker, Abdul Waheed, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. GPTAraEval: A comprehensive evaluation of ChatGPT on Arabic NLP. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 220–247, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Seonmin Koo, Chanjun Park, Jaehyung Seo, Seungjun Lee, Hyeonseok Moon, Jungseob Lee, and Heuiseok Lim. 2022. K-nct: Korean neural grammatical error correction gold-standard test set using novel error type classification criteria. *IEEE Access*, 10:118167–118175.

Ohjoon Kwon, Dohyun Kim, Soo-Ryeon Lee, Junyoung Choi, and SangKeun Lee. 2021. Handling out-of-vocabulary problem in hangeul word embeddings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3213–3221.

Myunghoon Lee, Hyeonho Shin, Dabin Lee, and Sung-Pil Choi. 2021. Korean grammatical error correction based on transformer with copying mechanisms and grammatical noise implantation methods. *Sensors*, 21(8).

Yongeun Lee. 2006. *Sub-syllabic constituency in Korean and English*. Ph.D. thesis, Northwestern University.

Junwei Liao, Sefik Eskimez, Liyang Lu, Yu Shi, Ming Gong, Linjun Shou, Hong Qu, and Michael Zeng. 2023. Improving readability for automatic speech recognition transcription. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–23.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational*

Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.

Natalia Lusin, Terri Peterson, Christine Sulewski, and Rizwana Zafer. 2023. Enrollments in languages other than english in us institutions of higher education: Fall 2021. *Modern Language Association of America*.

Jinjong Min, Sungjun Jung, Sehee Jung, Sungmin Yang, Junsang Cho, and Sunghwan Kim. 2020. Grammatical error correction models for korean language via pre-trained denoising. *Quantitative Bio-Science*, 39(1):17–24.

Roger Mitton. 1987. Spelling checkers, spelling correctors and the misspellings of poor spellers. *Information Processing Management*, 23(5):495–505.

Fayu Pan, Bin Cao, and Jing Fan. 2022. A multi-task learning framework for efficient grammatical error correction of textual messages in mobile communications. *EURASIP Journal on Wireless Communications and Networking*, 2022(1):99.

Er Sumreet Kaur Randhawa and Er Charanjiv Singh Saroa. 2014. Study of spell checking techniques and available spell checkers in regional languages: a survey. *International Journal For Technological Research In Engineering*, 2(3):148–151.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Felix Schneider, Sven Sickert, Phillip Brandes, Sophie Marshall, and Joachim Denzler. 2022. Metaphor detection for low resource languages: From zero-shot to few-shot learning in middle high german. In *Proceedings of the 18th Workshop on Multiword Expressions@ LREC2022*, pages 75–80.

Jae Jung Song. 2006. *The Korean language: Structure, use and context*. Routledge.

Linda Wiechetek, Flammie Pirinen, Mika Hämäläinen, and Chiara Argese. 2021. Rules ruling neural networks–neural vs. rule-based grammar checking for a low resource language. In *Proceedings of the International Conference Recent Advances In Natural Language Processing 2021*. INCOMA.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. Rethinking masked language modeling for Chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10743–10756, Toronto, Canada. Association for Computational Linguistics.

Soyoung Yoon, Sungjoon Park, Gyuwan Kim, Junhee Cho, Kihyo Park, Gyu Tae Kim, Minjoon Seo, and Alice Oh. 2023. Towards standardizing Korean grammatical error correction: Datasets and annotation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6713–6742, Toronto, Canada. Association for Computational Linguistics.

Damar Zaky and Ade Romadhony. 2019. An lstm-based spell checker for indonesian text. In *2019 international conference of advanced informatics: concepts, theory and applications (ICAICTA)*, pages 1–6. IEEE.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890, Online. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. 2022. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1244–1253, Dublin, Ireland. Association for Computational Linguistics.

A Typo Actions Details

We collected the NIKL Spelling Error Correction Corpus[6] designed for correcting spelling errors in text from websites. We derived statistical patterns of spelling errors from this dataset and designated each type as *Typo Actions*, further categorized into *Character/Jamo Actions*.

[6] https://corpus.korean.go.kr/

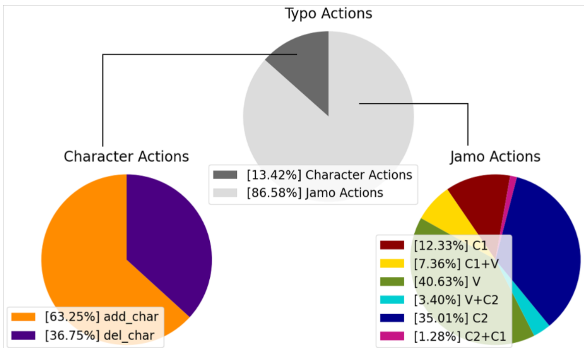

Figure 2: Statistical patterns of selecting each *Typo Action*. When one of the *Typo Actions* is initially selected, then one of the *Character/Jamo Actions* will be selected based on the following patterns.

Method	Word			Character			Average		
	P	R	F1	P	R	F1	P	R	F1
task description	62.00	62.00	61.95	**62.16**	**62.10**	**62.09**	62.08	62.05	62.02
task description + CoT	61.10	61.20	61.09	60.92	61.01	60.90	61.01	61.11	61.00
task description + CoT + ETA 1-shot	62.47	62.50	62.43	61.51	61.59	61.50	61.99	62.05	61.97
task description + CoT + ETA 4-shot	62.87	62.84	62.80	61.04	61.08	61.01	61.96	61.96	61.91
task description + CoT + ETA 8-shot	**63.13**	**63.04**	**63.02**	61.23	61.23	61.18	**62.18**	**62.14**	**62.10**

Table 4: Experimental results of correction the error corpus into an error-free corpus using the introduced spelling errors. When spelling errors were incorporated into both the test set and the few-shot samples, the P_{ensure} was set to 0.3.

Method	Word			Character			Average		
	P	R	F1	P	R	F1	P	R	F1
task description + CoT + ETA 1-shot	62.49	62.51	62.45	**61.54**	**61.61**	**61.52**	62.01	**62.06**	61.99
task description + CoT + ETA 4-shot	62.76	62.74	62.69	60.95	61.00	60.93	61.86	61.87	61.81
task description + CoT + ETA 8-shot	**63.03**	**62.95**	**62.93**	61.16	61.16	61.11	**62.09**	62.05	**62.02**

Table 5: Experimental results of correction the error corpus into an error-free corpus using the introduced spelling errors. When spelling errors were incorporated for the experiment, the test set had a P_{ensure} of 0.3 and the few-shot samples had a P_{ensure} of 0.6.

To determine which of these errors to generate, we measured the frequency of each type and conducted a statistical analysis of them, as illustrated in Figure 2. First, *Typo Actions* were divided into *Character/Jamo Actions*, allowing us to choose one of the two types. If *Character Actions* were selected, one of the sub-divided two types would be chosen, and if *Jamo Actions* were selected, one of the sub-divided six types would be chosen. This process applied information from the statistical pattern to determine the chosen type. If an error resulted from the final sound of a specific character, for example, *Jamo Actions* would be selected from the *Typo Actions* with a probability of 86.58%, and *C2* would be selected from the *Jamo Actions* with a probability of 35.01%.

B Prompt Design Details

We listed the actual texts used in each prompt. There are texts in place to prevent over-correction problem given the nature of the task, to support the reasoning, and to promote the utilization samples for few-shot learning.

- `text`$_{task}$: `Correct any errors in the following input written in Korean, while keeping the sentence unchanged as much as possible. Give me only the correct input, without any explanations.`

- `text`$_{cot}$: `You have to carefully check the input and correct any errors step by step.`

- `text`$_{n\text{-}shot}$: `Here is an example/are examples that you can refer to correct the given input.`

- `samples`: samples for few-shot learning selected from the train and validation sets.

We conducted 1, 4, and 8-shot learning, wherein the number of samples for the same input increased. We set the samples in the larger shots to encompass those in the smaller shots. For example, if sample *A* was selected in the 1-shot, the 4-shot samples include sample *A* along with new samples *B*, *C*, and *D*. Consequently, the 8-shot samples include samples *A*~*D* with new samples *E*, *F*, *G*, and *H*.

This was done to prevent performance from being solely determined by the random selection of additional samples. It allows for a quantitative comparison as the number of instances containing the introduced spelling errors increases with the growth of n, assuming the presence of the common samples for the same input.

C Experimental Details

C.1 Settings

We chose the `gpt-3.5-turbo-0125` model of ChatGPT. Depending on the nature of the task, to ensure the output focuses solely on spelling errors without generating excessive text that could lead to an over-correction problem, we configured the temperature and top_p to 0.1.

In our experiments, we conducted a single run when using only task description and CoT and two runs for few-shot learning. The average performance of each result was reported, with independent samples utilized for the few-shot learning.

C.2 Metrics

We employed precision, recall, and f1-score for evaluation. In contrast to other downstream tasks in text generation, spell checking does not involve generating new tokens; instead, the goal is to correct errors while maintaining the correct words. Therefore, we devised the metrics evaluating the correctness and order of words between the outputs and gold texts, as well as the correctness and order of characters.

C.3 Additional Experiments

We further experimented with the more challenging test set with lower values of P_{ensure}, and the results are presented in Table 4~5. We kept the value of P_{ensure} of 0.6 to the test set and varied its value to the samples for few-shot learning in Table 1~2. In this section, we conducted experiments using the same samples for few-shot learning, while applying P_{ensure} of 0.3 to the test set.

The results at the word-level exhibited a gradual improvement in performance with an increasing number of samples with few-shot learning. At the character-level, a slight improvement was observed when relying solely on the task description, and as a result, the best averaged performance was obtained through few-shot learning with the introduced spelling errors. In terms of the performance variation, There was a slight advantage with a P_{ensure} of 0.3 compared to 0.6 on the test set. This indicated that correcting the introduced spelling errors through prompting performed well on more challenging input. However, further experiments with various values of P_{ensure} are needed for conclusive results. It is important to note that we used P_{ensure} values of 0.6 and 0.3 throughout all experiments, but this choice was based on quantitative comparisons, and the users have the flexibility to adjust the value as desired.