Rich Semantic Knowledge Enhanced Large Language Models for Few-shot Chinese Spell Checking

Ming Dong[1,2,3], **Yujing Chen**[1,2,3], **Miao Zhang**[4], **Hao Sun**[1,2,3], **Tingting He**[1,2,3,*]
[1]Hubei Provincial Key Laboratory of Artificial Intelligence and Smart Learning,
[2]National Language Resources Monitoring and Research Center for Network Media,
[3]School of Computer, Central China Normal University, Wuhan, China
[4]School of Computer Science and Information Engineering, Hubei University, Wuhan, China
{dongming,haosun,tthe}@ccnu.edu.cn
{yujingchen}@mails.ccnu.edu.cn
{miaozhang}@hubu.edu.cn

Abstract

Chinese Spell Checking (CSC) is a widely used technology, which plays a vital role in speech to text (STT) and optical character recognition (OCR). Most of the existing CSC approaches relying on BERT architecture achieve excellent performance. However, limited by the scale of the foundation model, BERT-based method does not work well in few-shot scenarios, showing certain limitations in practical applications. In this paper, we explore using an in-context learning method named RS-LLM (**R**ich **S**emantic based LLMs) to introduce large language models (LLMs) as the foundation model. Besides, we study the impact of introducing various Chinese rich semantic information in our framework. We found that by introducing a small number of specific Chinese rich semantic structures, LLMs achieve better performance than most of the BERT-based model on few-shot CSC task. Furthermore, we conduct experiments on multiple datasets, and the experimental results verified the superiority of our proposed framework.

1 Introduction

Spell checking (SC) aims to utilize intelligent methods to automatically identify and correct errors in text. This technology facilitates nature language processing applications to correct the errors from different text input systems, such as speech to text (STT) and optical character recognition (OCR). In recent years, SC has attracted tremendous attention from the research community (Chodorow et al., 2007; Malmi et al., 2019; Mallinson et al., 2020). Chinese spell checking (CSC) specifically refers to SC for Chinese text. Compared with the relatively complete SC technology, CSC still cannot be perfectly applied to various practical scenarios, and there are still many problems that need to be solved (Zhang et al., 2023).

As an ideogram, the usage and structure of Chinese are very different from English, which leads to different challenges for CSC and SC. First of all, the pronunciation of Chinese varies greatly, and it is difficult to easily infer the glyphs from the pronunciation. When listening to a piece of Chinese speech, if you do not understand the context, the text result obtained based on the speech is likely to contain a large number of homophones. In addition, the glyph structure of Chinese is more diverse, resulting in more types of errors. Therefore, CSC task mainly needs to handle two types of error texts. One is a text based on the STT system, which contains a large number of homophonic errors. The other is text generated by OCR-based systems that mainly contains glyph errors. To address these two types of spelling errors, most of the existing studies use models based on BERT architecture, then introducing the external glyph-phonetic features (Ji et al., 2021; Xu et al., 2021; Ji et al., 2021).

In practical Internet applications, the catchwords from hot topics vary rapidly and unlabeled incorrect sentences emerge constantly, resulting in few-shot scene for CSC. However, existing BERT-based models are difficult to be conduct in few-shot scene because of the limited scale of the foundation model. Large language models (LLMs) show remarkable ability on semantic analyzing, positioning them to become an optimal foundation model for CSC. This paper focus on CSC in few-shot scene. We build a Chinese rich semantic corpus (See details in Section 4.1). Besides, we choose LLMs as foundation and integrate Chinese rich semantic knowledge by in-context learning. Furthermore, we conduct experiments on several datasets. The contributions of this study are summarized as follows:

- We propose an in-context learning based method to introduce LLMs to CSC task, which improves the performance of few-shot

[*] Corresponding author.

August 11-16, 2024 ©2024 Association for Computational Linguistics

scenarios.

- We propose the paradigm of prompt template designing for CSC.
- We conduct experiment to compare different Chinese Rich Semantic structures. And we find the best structures for LLMs based CSC tasks.

2 Related Work

Due to the lack of parallel corpus training data, early CSC methods mainly rely on linguistic knowledge to manually design rule-based methods (Mangu and Brill, 1997; Jiang et al., 2012). Subsequently, machine learning models become the main paradigm for CSC tasks (Chen et al., 2013; Yu and Li, 2014). Machine learning typically employs language models, such as n-grams, to detect error locations. Then uses confusion sets and character similarities to correct potential misspelled characters and candidate correct characters, and finally scores replace sentences through the language model to determine the best correction solution (Liu et al., 2013; Xie et al., 2015).

The field of CSC advances significantly with the development of deep learning, particularly through pre-trained models like BERT (Devlin et al., 2019). Pre-trained models such as BERT are known for their context awareness and transfer learning capabilities. Most current CSC models with better performance use BERT as the baseline model. Hong et al. (2019) innovates by modeling CSC as a BERT token classification task, utilizing the Confidence-Character Similarity Decoder (CSD). Zhang et al. (2020) enhances this approach by combining error identification and correction losses with a soft mask strategy. Addressing the prevalent issue of phonetic and glyph similarities in spelling errors, the integration of these features with semantic information is now a primary research focus. Liu et al. (2021) suggests incorporating confusion sets into pre-training with a GRU network to better mimic real errors and model character sound-shape similarities. Xu et al. (2021) proposes a multi-modal approach to capture semantic, phonetic, and graphical information and the use of adaptive gating modules to merge semantic, phonetic, and glyph features in CSC. Ji et al. (2021) introduces SpellBERT, which integrates radical features into character representation using a graph convolution network. The SCOPE model (Li et al., 2022) further develops this field by adding pronunciation prediction tasks in training, forging a deeper connection between semantics and phonetics, and employing iterative reasoning strategies to bolster CSC model performance.

Recently, the development and advancement of LLMs have brought natural language processing to the next stage. Li et al. (2023b) analyzes the correction ability of the OpenAI[1]'s existing LLMs and finds that they still fall short of the CSC capabilities of previous fine-tuned models.

3 Preliminary

3.1 Chinese Rich Semantics

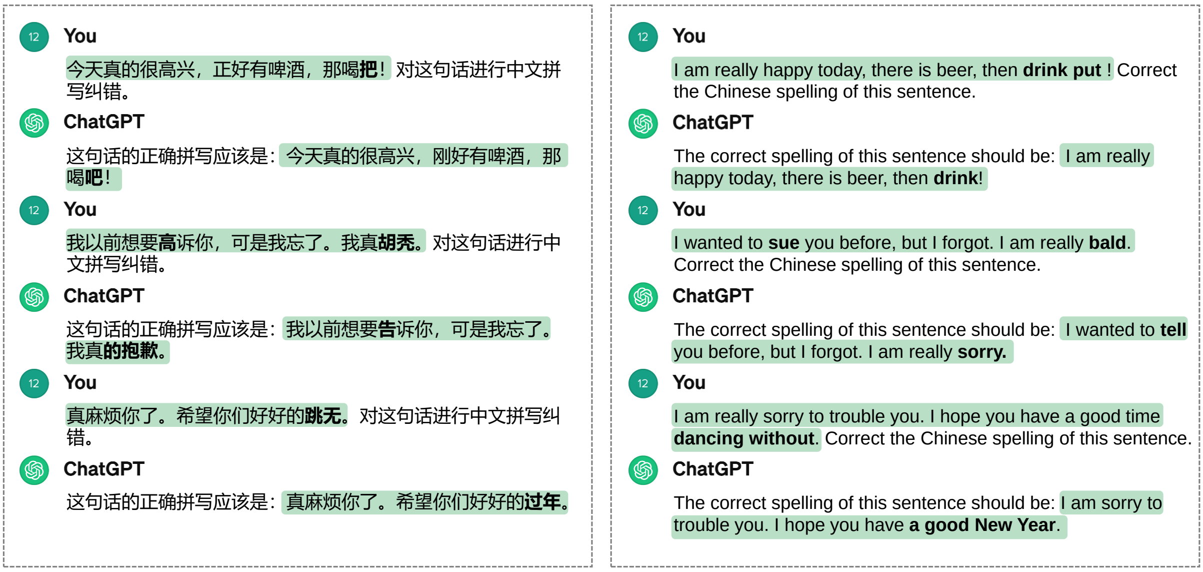
Chinese is the carrier of the inheritance and development of Chinese civilization. As a hieroglyphic script that has lasted for thousands of years, Chinese characters have rich semantic structures, including traditional characters, glyphs, phonetic, tones and other features. Chinese contains a large number of phonograms, whose pronunciation and meaning are contained in radicals. For example, "海 (sea)", "河 (river)" and "湖 (lake)" all have the radical " 氵 ", related to water (Sun et al., 2021). For another example, "株 (zh ū)", "诛 (zh ū)" and "珠 (zh ū)" all have the same phonetic tone "朱 (zh ū)", so the pronunciation of these characters is similar. In essence, radicals can be regarded as a kind of classification label information. This classification information serves as traditional context. The additional supplement of semantic features has important semantic value. In addition to the features at the glyph level, Chinese also has features such as morpheme diversity, ambiguity, and structural diversity.

3.2 Task Definition

Formulate definition of CSC: CSC focuses on identifying and rectifying spelling errors in a given text sequence. Essentially, it involves processing a sequence of characters $X = \{x_1, x_2, ..., x_n\}$ and producing a corrected sequence $Y = \{y_1, y_2, ..., y_n\}$, where n is the number of word. Here, X denotes the initial text with potential errors, and Y signifies the amended, correct text. The two sequences X and Y have the same length.

Few-shot Learning on CSC: Few-shot learning refers to provide a model named L with K pairs of contexts and corresponding required answers as examples, and then a context that requires model L to reason. The goal of model L is to generate

[1] https://openai.com/



You

今天真的很高兴，正好有啤酒，那喝**把**！对这句话进行中文拼写纠错。

ChatGPT

这句话的正确拼写应该是：今天真的很高兴，刚好有啤酒，那喝**吧**！

You

我以前想要**高**诉你，可是我忘了。我真**胡秃**。对这句话进行中文拼写纠错。

ChatGPT

这句话的正确拼写应该是：我以前想要**告**诉你，可是我忘了。我真**的抱歉**。

You

真麻烦你了。希望你们好好的**跳无**。对这句话进行中文拼写纠错。

ChatGPT

这句话的正确拼写应该是：真麻烦你了。希望你们好好的**过年**。

You

I am really happy today, there is beer, then **drink put** ! Correct the Chinese spelling of this sentence.

ChatGPT

The correct spelling of this sentence should be: I am really happy today, there is beer, then **drink**!

You

I wanted to **sue** you before, but I forgot. I am really **bald**. Correct the Chinese spelling of this sentence.

ChatGPT

The correct spelling of this sentence should be: I wanted to **tell** you before, but I forgot. I am really **sorry.**

You

I am really sorry to trouble you. I hope you have a good time **dancing without**. Correct the Chinese spelling of this sentence.

ChatGPT

The correct spelling of this sentence should be: I am sorry to trouble you. I hope you have **a good** New **Year**.

Figure 1: The performce of CSC tasks using LLM (ChatGPT) without specific restrictions on input prompts.

an appropriate answer for the last context. During inference, model L is guided by K context-answer pairs without any updates to its parameters (Brown et al., 2020).

In few-shot learning for CSC task, we set K to 3. Then we have a few-shot learning prompt P, which contains three sentences pairs:

$$P = \{p_1, p_2, p_3\} \tag{1}$$

where p_i contains an incorrect sentence and its corresponding correct sentence.

Then the sentence X that need to be spell checked:

$$X = \{x_1, x_2, \ldots, x_n\} \tag{2}$$

where x_i denotes a word of sentence X. n is the length of sentence X.

For model L, the inputs are P and X, and the output is the correct sentence Y corresponding to the sentence X.

$$Y = L(P, X) \tag{3}$$

$$Y = \{y_1, y_2, \ldots, y_n\} \tag{4}$$

where y_i denotes a word of sentence Y and n is the length of sentence Y.

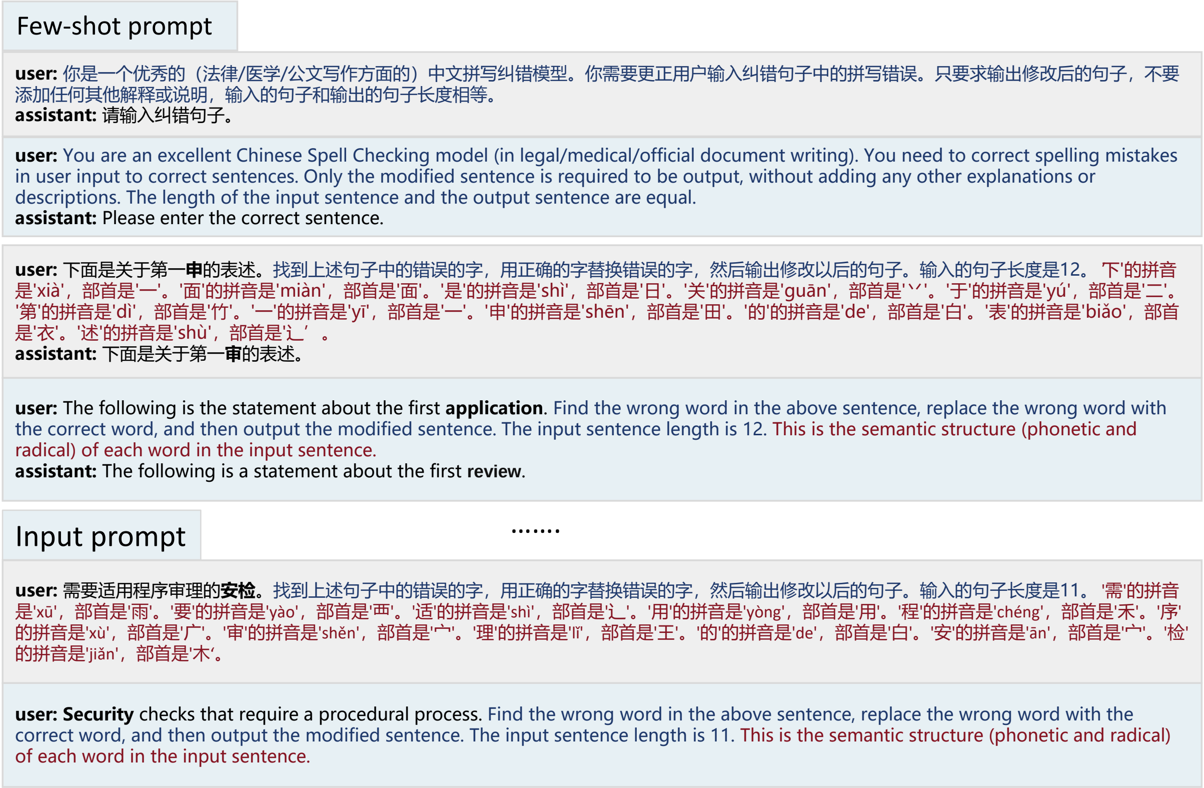
3.3 Difficulties of LLMs based CSC Tasks

The versatility of LLMs gives them significant text polishing capabilities. Since there are no specific restrictions on input prompts, LLMs tend to perform freely in CSC tasks. However, free play of LLMs may result in LLMs outputting sentences that are completely grammatically correct. However, the sentences output by LLMs are different from the standards established by existing CSC datasets and evaluation indicators. Therefore, these existing traditional datasets are used without specific restrictions on input prompts. It becomes challenging to objectively and realistically evaluate the spell checking performance of LLMs. An example of using LLMs (ChatGPT) to perform a CSC task without setting specific restrictions on input prompts is shown in Fig.1. From Fig.1, it can be observed that two main problems are prone to occur in LLMs when performing CSC tasks, one is the length of the input sentence and the output sentence are inconsistent. For instance, the sentence "I am really bald (我真胡秃)", which should be corrected to "I am really confused (我真糊涂)". However, LLMs correct this sentence to "I am really sorry (我真的抱歉)". Another problem is that LLMs easily rewrite the input sentences semantically. For example, "I hope you have a good time dancing without (希望你们好好的跳无)", in which "dancing without (跳无)" should be corrected to "dancing (跳舞)". Instead, LLMs correct this sentence to "I hope you have a good New Year (希望你们好好的过年)".

4 Method

4.1 Chinese Rich Semantic Corpus

Chinese, a logo-graphic language, inherently possesses a rich semantic depth in its character glyph,



Few-shot prompt

user: 你是一个优秀的（法律/医学/公文写作方面的）中文拼写纠错模型。你需要更正用户输入纠错句子中的拼写错误。只要求输出修改后的句子，不要添加任何其他解释或说明，输入的句子和输出的句子长度相等。
assistant: 请输入纠错句子。

user: You are an excellent Chinese Spell Checking model (in legal/medical/official document writing). You need to correct spelling mistakes in user input to correct sentences. Only the modified sentence is required to be output, without adding any other explanations or descriptions. The length of the input sentence and the output sentence are equal.
assistant: Please enter the correct sentence.

user: 下面是关于第一**申**的表述。找到上述句子中的错误的字，用正确的字替换错误的字，然后输出修改以后的句子。输入的句子长度是12。 '下'的拼音是'xià'，部首是'一'。'面'的拼音是'miàn'，部首是'面'。'是'的拼音是'shì'，部首是'日'。'关'的拼音是'guān'，部首是'丷'。'于'的拼音是'yú'，部首是'二'。'第'的拼音是'dì'，部首是'竹'。'一'的拼音是'yī'，部首是'一'。'申'的拼音是'shēn'，部首是'田'。'的'的拼音是'de'，部首是'白'。'表'的拼音是'biǎo'，部首是'衣'。'述'的拼音是'shù'，部首是'辶' 。
assistant: 下面是关于第一**审**的表述。

user: The following is the statement about the first **application**. Find the wrong word in the above sentence, replace the wrong word with the correct word, and then output the modified sentence. The input sentence length is 12. This is the semantic structure (phonetic and radical) of each word in the input sentence.
assistant: The following is a statement about the first **review**.

Input prompt

....

user: 需要适用程序审理的**安检**。找到上述句子中的错误的字，用正确的字替换错误的字，然后输出修改以后的句子。输入的句子长度是11。 '需'的拼音是'xū'，部首是'雨'。'要'的拼音是'yào'，部首是'覀'。'适'的拼音是'shì'，部首是'辶'。'用'的拼音是'yòng'，部首是'用'。'程'的拼音是'chéng'，部首是'禾'。'序'的拼音是'xù'，部首是'广'。'审'的拼音是'shěn'，部首是'宀'。'理'的拼音是'lǐ'，部首是'王'。'的'的拼音是'de'，部首是'白'。'安'的拼音是'ān'，部首是'宀'。'检'的拼音是'jiǎn'，部首是'木'。

user: Security checks that require a procedural process. Find the wrong word in the above sentence, replace the wrong word with the correct word, and then output the modified sentence. The input sentence length is 11. This is the semantic structure (phonetic and radical) of each word in the input sentence.

Figure 2: Task-specific few-shot prompts for CSC tasks. We marked the semantic structure information (speech and radicals) and key information related to the task features in the prompts in different color.

potentially enhancing the expressiveness of LLMs. Our work focuses on the GB2312 simplified Chinese coding table, a standard set by China's State Administration of Standards on May 1, 1981[2]. This table consists of 6,763 Chinese characters, divided into 3,755 primary and 3,008 secondary characters. It contains the most commonly used Chinese characters. In order to obtain the detailed information of this coding table, we collected various basic attributes of each Chinese word, such as its *phonetic*, *radical*, *phonetic notation*, *strokes*, *outside strokes (the strokes except the radical)*, *stroke order*, *structure*, *Unicode*, *Wubi code*, *Cangjie code*, *Zheng code*, *Four-corner code*, as well as *glyph images from different historical periods*. Despite the large amount of data collected, we note issues with the quality and completeness of the data. To address these issues, we manually annotate the collected information to ensure a more accurate and comprehensive dataset. We give the attributes of the word 海 (sea) in the dataset as shown in Fig.3. In order to understand and better utilize these properties, we classify these features into the following three categories:

- **Phonetic Features**: *Phonetic* uses Latin letters to represent the pronunciation of Chinese characters. *Phonetic notation* is a phonetic system that uses symbols to represent the speech of Chinese characters. *Zheng Code* is a Chinese character input method that assigns codes based on the initials of the phonetic pronunciation.
- **Glyph Features**: *Radical* is a category according to the type and side of the Chinese characters, and all the Chinese characters are bound to be classified in a certain radical. *Strokes* refers to the number of lines needed to write Chinese character. *Outside strokes* mean the number of lines needed to write Chinese character except the radical. *Structure* refers to the internal organization of Chinese characters, including the arrangement of radicals, strokes, and components.
- **Input Coding Features**: *Stroke Order* indicates the sequence in which strokes are written when forming a Chinese character. Proper stroke order is important for correct character writing. *Cangjie code* assigns codes to characters based on their shapes and compo-

[2] https://openstd.samr.gov.cn/bzgk/gb

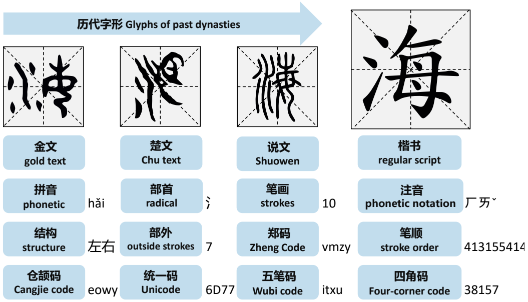


Figure 3: Various attributes of the word 海 (sea) in the Chinese Rich Semantics.

nents. *Unicode* is an international standard for character encoding that assigns unique codes to characters from various writing systems, including Chinese characters. *Wubi code* assigns codes based on the five basic strokes. *Four-Corner code* is a Chinese character input method that assigns codes based on the shapes of the four corners of a character.

We publish the information we collect at dropbox. [3]

4.2 In Context Learning for CSC

4.2.1 Motivation

A variety of studies (Xie et al., 2022; Dai et al., 2023; Bansal et al., 2023) have revealed that LLMs exhibit exceptional in-context learning capabilities (Dong et al., 2023). In-context learning can quickly improve the task-specific performance of LLMs by providing a limited set of task-related examples, which can quickly adapt to most LLMs without the need for separate training for each LLM. As mentioned in Section 3.3, when LLMs perform CSC tasks without specific task constraints on input prompts, the answers generated by LLMs are very likely to be inconsistent with the existing evaluation indicators of CSC tasks. Therefore, in order to accurately and objectively explore the performance ability of LLMs on CSC tasks, we design the task-specific prompts as shown in Fig.2.

4.2.2 Prompt Design

First of all, we give the identity and task description in the prompt. According to Li et al. (2023a), we know that Role Attribution (“You are an excellent Chinese Spell Checking model") can effectively stimulate LLMs’ comprehension. The task description requires the LLMs to only correct spelling mistakes, thus limiting the LLMs’ semantic rephrasing of the input sentence. Since the evaluation of CSC task requires the input sentence and the output sentence to be same length, the LLMs are required not to add any other explanations and descriptions of output, so as to ensure the length remains unchanged.

Then, we give three pairs of input sentences and their corrected sentences as examples, carefully selected from the corresponding training set of the dataset we use. From Liu et al. (2010), Chinese text errors are primarily caused by characters that are visually and phonetically similar. These three sentence pairs contain a set of phonetic errors, a set of glyph errors, and a set of correct sentences that do not need to be corrected.

Next, we add specific task requirements to the input sentences of three pairs of sentences, which is different from the foremost task description. The foremost task description requires LLMs to correct the spelling errors of the input sentences. We concretize the task here, asking LLMs to find spelling errors in sentences and replace wrong word with correct one. This allows LLMs to have a clearer comprehension of CSC task. We append the length of the input sentence to the end to indicate the LLMs that we limit the length of the output sentence in the first prompt. We also include the phonetic and radical information of each Chinese character in the input sentence, drawn from Chinese Rich Semantic Corpus outlined in Section 4.1. It is hoped that by adding this information to the prompt, the LLMs can strengthen their understanding of the input sentence. Therefore, LLMs can better correct phonetic and visual errors that may occur in the sentences. These are in the form of four pairs of historical dialogues that form a few-shot prompt input to LLMs, hoping to stimulate the in-context learning capabilities of LLMs at once.

Considering the length limit of LLMs on the length of the input (including historical dialogue), and the deterioration of the semantic understanding of LLMs with long input lengths. For each sentence to be corrected in the test set, we clear the historical memory of the LLMs and add our few-shot prompt.

4.3 Introspection Mechanism

A significant challenge with using LLMs for spell checking is that LLMs tend to over-modify and arbitrarily change sentence lengths. In order to avoid the impact of this change on the evaluation indicators, we use a Introspection mechanism.

[3] https://www.dropbox.com/scl/fo/0r1jw4l1ex3w0lyojsfpf/AGVMdlOFpwDqoOFrxqWXJko?rlkey=18wke1vmhj6muwufvu3aazji2&st=vgh1drbi&dl=0

Specifically, after LLMs generate the correction sentence, we send this answer and the original input sentence to LLMs again. LLMs are required to introspect two questions: 1) Whether the lengths of the two sentences are consistent. 2) Whether the added rich semantic information is effectively used in this error correction process. Only when the answers to both questions are "yes" will the answer be output as the final error correction result. Otherwise, the current conversation will be added to the historical conversation and provided as context to the model. In the conversation, it indicates that the sentence length in this answer does not match the input sentence length or the semantic information is not used, and then sends a reply request to LLMs again. In the introspection mechanism of this experiment, we set the maximum number of loops to 5. If the answer cannot be obtained after five requests to LLMs, it is judged that LLMs cannot correct the sentence and uses original input sentence without introspection as the answer. And consider that the model's ability to understand long context will deteriorate. We only add the latest round of dialogue in each loop within the context of the original design.

5 Experiments

5.1 Datasets and Evaluation Metrics

Datasets: To evaluate the effectiveness of our proposed method, we choose two widely used datasets. The first, SIGHAN15 (Tseng et al., 2015), consists of handwritten samples from learners of Chinese as a Second Language (CSL). These samples provide a rich source of real-world language usage by non-native speakers, offering insights into common errors and patterns in learning Chinese. The second dataset (Lv et al., 2023) is specialized and segmented into three distinct domains to provide a more comprehensive understanding of language use in specific contexts. For the LAW domain, data is sourced from the stems and choices of multiple-choice questions in judicial examinations, reflecting the formal and technical language of the legal field. The MED domain encompasses data from question-and-answer pairs drawn from online medical consultations, showcasing the specific terminology and communication style in healthcare. The ODW (Official Document Writing) domain includes data from various official documents such as news, policies, and state reports on national conditions, representing formal and structured writing styles. The statistics of the test data from these four datasets we used are shown in Table 1.

Test Data	#Sent	Avg. Length	#Errors
SIGHAN15	1100	30.6	703
LAW	500	29.7	356
MED	500	49.6	345
ODW	500	40.5	403

Table 1: Statistical information regarding the dataset in our experiments, which includes the total count of sentences (#Sent), the average length of these sentences (Avg. Length), and the total number of spelling mistakes (#Errors).

Evaluation Metrics: By following the existing work (Xu et al., 2021; Lv et al., 2023), we evaluate the performance on two metrics: detection and correction. For detection, a sentence is considered correct if it successfully identifies all spelling errors. For correction, the model not only identifies but also rectifies all erroneous characters by replacing them with the correct ones. We provide accuracy, precision, recall, and F1-scores for both metrics.

5.2 Baselines

We choose two widely used LLMs as foundational models.

gpt-3.5-turbo[4]: One of a series of LLMs provided by OpenAI, which can be accessed through the API. The gpt-3.5-turbo serves as the underlying module for ChatGPT and is trained on GPT-3 using Reinforcement Learning from Human Feedback (RLHF).

ChatGLM2-6B[5]: An open-source bilingual (Chinese-English) chat model. ChatGLM2-6B employs a GLM-based (Du et al., 2022) hybrid objective function and has been pre-trained on 1.4 trillion bilingual tokens and human preference alignment training.

We choose several advanced CSC models as baselines:

BERT (Devlin et al., 2019): BERT encodes the input sentence to get semantic information, followed by using a classifier to select the correct character from the vocabulary.

ChineseBERT (Sun et al., 2021): ChineseBERT encodes the input sentence to get semantic,

[4] https://platform.openai.com/docs/api

[5] https://huggingface.co/THUDM/chatglm2-6b

Data	without prompt	with prompt
SIGHAN15	578	724
LAW	83	125
MED	237	262
ODW	294	347

Table 2: The number of sentences whose length does not change using specific prompts and not using specific prompts on different data sets. Experimental result statistics are based on gpt-3.5-turbo.

phonetic, and graphical information, then use a classifier to select the correct character from the vocabulary.

ReaLiSe (Xu et al., 2021): This CSC model captures semantic, phonetic, and graphical information of input characters using multimodality for prediction.

Scope (Li et al., 2022): This CSC model introduces an auxiliary task of Chinese pronunciation prediction (CPP) to improve CSC task.

5.3 Implementation Details

For all methods, the settings of hyper-parameters follow the optimal parameters in the open source code corresponding to the model. In order to achieve few-shot scenarios, all contrast experiments randomly selected 10 samples of data from the training set, trained for 1000 epochs, and then selected the best performance. The LLMs experiment uses the same few-shot prompt designed for CSC task as RS-LLMs, but does not add semantic information. All experiments were conducted on a RTX-4090 with 24G memory.

5.4 Experimental Results

Table 3 shows the evaluation results of our RS-LLMs comparing to other models on the test sets of SIGHAN15, LAW, MED, and ODW. It is observed that our RS-LLM consistently outperforms all baselines on all datasets in all metrics. Especially compared with LLM, LLM uses the same few-shot prompt as RS-LLM but does not add semantic information when performing CSC tasks. The experimental results of this period verify the effectiveness and superiority of our semantic information on LLM for CSC task.

From time to time experiments, we found that the experimental results are not completely consistent because the performance of online APIs varies. Therefore, we use ± in the results to explain the error band of the experiment.

As shown in Table 3, our spell-checking approach, RS-LLM on ChatGLM2-6B, achieves a notable 8.0% improvement in error detection and an 9.8% increase in correction on SIGHAN 15 using Rich Semantic, compared to the standard LLM. Impressively, RS-LLM on gpt-3.5-turbo registers a 9.9% boost in F1-score for detection and a 12.3% leap in correction. In the LAW dataset, RS-LLM on ChatGLM2-6B beats ChatGLM2-6B by 5.8% in detection and 6.6% in correction, while RS-LLM on gpt-3.5-turbo outperforms gpt-3.5-turbo by 16.9% and 14.6%, respectively. The trend continues in the MED dataset, where RS-LLM on ChatGLM2-6B surpasses ChatGLM2-6B by 13.1% in detection and 9.3% in correction, and RS-LLM on gpt-3.5-turbo exceeds gpt-3.5-turbo by 15.3% and 13.7%. On the ODW dataset, our method also shows significant gains, with RS-LLM on ChatGLM2-6B leading ChatGLM2-6B by 6.9% in detection and 5.6% in correction, and RS-LLM on gpt-3.5-turbo outdoing gpt-3.5-turbo by 14.1% and 12.7%.

5.5 Analyses and Discussions

In order to verify the effectiveness of the prompt sentence, we count the number of sentences whose length remained the same with and without a specific prompt, and the experimental results are shown in Table 2. It is obvious that LLM performs better than most of the BERT-based CSC models at both the detection and correction levels. The ability of the BERT-based CSC model to detect and correct erroneous characters is overly dependent on the training data of CSC, especially the ability to correct erroneous characters. The LLMs based approach shows better generalization in the few-shot scenario. However, we find that LLM's performance on the CSC task heavily depends on the foundation model. When the scale of parameters of the foundation model is larger, the model performs better on the CSC task. Although the performance of RS-LLM cannot currently outperform Scope, we believe that with the continuous update of the base model performance, the RS-LLM method will continue to improve and show a better improvement trend.

5.5.1 Impact of Different In-context Learning Approaches

Fig.4 and Fig.5 show the evaluation results of different in-context learning approaches. From Fig.4

Dataset	Method	Detection Level				Correction Level			
		Acc.	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**
SIGHAN15	BERT	17.8	15.1	16.1	15.8	15.4	3.6	2.4	2.8
	ChineseBERT	16.2	10.4	12.9	11.5	13.2	2.7	5.5	3.6
	ReaLiSe	27.5	16.2	18.6	17.3	22.3	3.6	3.1	3.4
	Scope	64.7	61.7	34.0	43.9	58.1	37.2	20.5	26.5
	ChatGLM2-6B	18.3	7.1	10.1	8.3±1.1	17.0	3.3	5.3	4.1±2.8
	RS-ChatGLM2-6B	30.7	14.9	18.0	16.3±2.3	28.6	12.4	15.6	13.9±3.9
	gpt-3.5-turbo	36.4	22.3	33.0	26.6±3.1	34.3	19.3	28.6	23.1±3.5
	RS-gpt-3.5-turbo	**50.6**	**32.5**	**41.6**	**36.5±4.7**	**48.1**	**31.2**	**40.8**	**35.4±5.9**
LAW	BERT	14.4	8.8	12.3	10.3	12.7	1.35	0.4	0.6
	ChineseBERT	15.0	12.3	14.3	13.2	13.5	0.8	1.6	1.1
	ReaLiSe	23.3	15.4	18.2	16.7	20.2	1.6	1.5	1.5
	Scope	66.2	50.4	48.6	49.5	58.6	35.0	33.7	34.3
	ChatGLM2-6B	36.6	18.5	25.5	21.4±2.7	34.8	15.9	21.9	18.5±3.3
	RS-ChatGLM2-6B	45.2	24.2	25.1	24.6±3.4	40.4	22.7	24.8	23.7±3.2
	gpt-3.5-turbo	48.8	30.2	38.8	34.0±3.6	46.2	26.2	33.7	29.5±3.6
	RS-gpt-3.5-turbo	**64.6**	**46.6**	**56.1**	**50.9±9.4**	**60.8**	**40.4**	**48.6**	**44.1±8.7**
MED	BERT	14.6	7.1	13.3	9.2	12.6	1.8	0.4	0.6
	ChineseBERT	14.0	8.1	10.4	9.1	11.2	0.6	1.5	0.9
	ReaLiSe	26.2	9.7	16.8	12.3	18.6	1.5	0.9	1.1
	Scope	66.4	45.6	53.9	49.4	56.5	27.3	32.3	29.6
	ChatGLM2-6B	31.2	15.7	23.0	18.7±5.6	28.4	14.7	19.2	16.7±6.5
	RS-ChatGLM2-6B	45.2	30.4	33.3	31.8±6.7	39.9	22.6	30.6	26.0±4.2
	gpt-3.5-turbo	41.4	16.8	30.1	21.5±1.2	38.6	13.3	23.9	17.1±2.7
	RS-gpt-3.5-turbo	**56.0**	**31.0**	**45.2**	**36.8±8.7**	**43.3**	**25.3**	**39.3**	**30.8±6.8**
ODW	BERT	16.8	13.2	15.6	14.3	13.1	4.4	1.5	2.3
	ChineseBERT	15.9	12.2	14.2	13.1	11.6	2.0	3.8	2.7
	ReaLiSe	30.2	18.5	26.8	21.9	25.4	5.4	4.2	4.7
	Scope	75.0	65.5	58.7	62.0	70.8	56.5	50.7	53.5
	ChatGLM2-6B	47.6	28.6	32.5	30.4±6.5	44.2	27.0	31.1	28.9±7.8
	RS-ChatGLM2-6B	56.4	32.1	37.8	34.7±2.7	50.8	29.8	32.4	31.0±3.2
	gpt-3.5-turbo	63.0	45.8	50.0	47.8±8.5	59.2	39.2	42.8	40.9±7.9
	RS-gpt-3.5-turbo	**72.4**	**59.1**	**64.9**	**61.9±2.6**	**70.2**	**50.2**	**57.6**	**53.6±5.1**

Table 3: The performance of all baselines and RS-LLMs. RS-gpt-3.5-turbo means RS-LLM on gpt-3.5-turbo and RS-ChatGLM2-6B means RS-LLM on ChatGLM2-6B. ChatGLM2-6B and gpt-3.5-turbo only utilize identical few-shot prompts as RS-LLM, without semantic prompt and introspection.**The bold information** indicates the best results except Scope (Scope is the best BERT-based model in few-shot scene), and ± indicates the error band of the results.

and Fig.5, we discover that the performance of CSC task on LLMs improves as the number of examples increasing, both in terms of detection and correction. Additionally, it's clear that RS-LLM is more effective than most BERT-based models in terms of zero-shot, one-shot, and few-shot scene. Such experimental results reflect the effectiveness of our in-context learning strategy designed for CSC task. The specific experimental results are shown in Table 5 in Appendix A.

5.5.2 Impact of Different Chinese Rich Semantic Information

In order to further study the impact of semantic information on LLMs when performing CSC tasks. We design to use phonetic information, radical information, structural information and strokes information. Each of these four types of information is added to our zero-shot prompt, one-shot prompt, and few-shot prompt. To explore the impact of these four prompts on the test set of SIGHAN15, LAW, MED, and ODW. The specific experimental results are Table 4 in Appendix A.

The experimental results show that individual phonetic, radical, structural, and stroke information leads to improvements in CSC task across all datasets. Notably, the phonetic and radical information contribute the most significant enhancements, followed by structural information. While strokes information does show some improvement, it's relatively puny compared to the others.

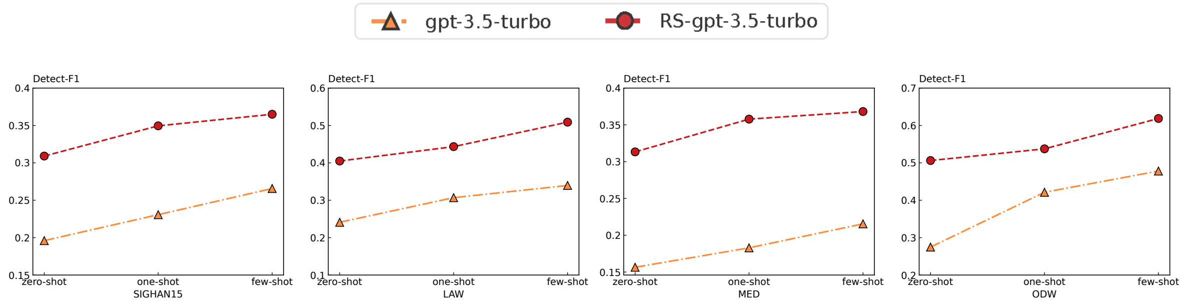


Figure 4: Detect-F1 score trends on the test dataset.

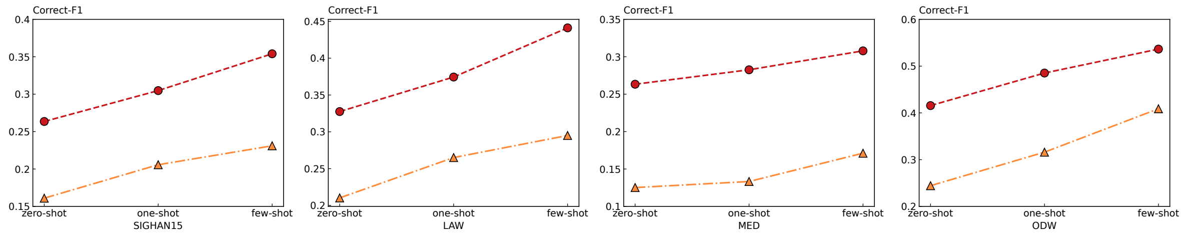


Figure 5: The impact of different number of prompt samples.

6 Conclusion and Future Work

In this paper, we introduce an in-context learning method named RS-LLM for CSC task, one of whose core components is using Chinese Rich Semantics in LLMs for CSC task. RS-LLM utilizes adding a small number of specific Chinese rich semantic information into a specific few-shot prompt set for CSC task, aiming to allow LLMs to have a fuller understanding of the semantics when doing CSC task. Experimental results show that this LLMs-independent approach can help existing LLMs better recognize and correct phonetically and visually erroneous characters in CSC tasks. Considering the impact of the similarity of errors in the few-shot prompt and errors in the sentence on LLMs' understanding of the sentence when LLMs perform CSC task. In the future, we will try to construct dynamic prompt for each sample through semantic similarity retrieval.

Limitations

To verify the effectiveness of RS-LLM, we conducted extensive experiments on two benchmark datasets of different domains and scales. The results indicate that RS-LLM delivers SOTA results in few-shot scenarios. Since most of the errors in the CSC dataset are attributed to visual and phonetical errors, we incorporate phonetic and radical information into the prompt template. However, it is difficult to ensure that the existing manually formulated prompt templates are optimal, and the optimal prompt sentences for CSC require further research. Furthermore, relevant examples have been carefully selected to enable LLMs to identify potential visual and speech errors in a small number of scenarios. No consideration was given to the ability to motivate the LLMs through semantic similarity. We recognize these two limitations and plan to address them in future research efforts.

Ethics Statement

We adhere to and advocate for the principles outlined in the ACL Code of Ethics. The primary focus of this paper is on the task of CSC, with an objective to enhance the performance of LLMs in this task by incorporating semantic knowledge into the template. The datasets utilized in our research are obtained from openly published sources, ensuring they are free from privacy or ethical concerns. In our approach, we consciously avoid introducing or magnifying any social or ethical biases in the model or data. Consequently, we anticipate no direct social or ethical challenges as a result of our research.

Acknowledgments

This work was partly supported by China Postdoctoral Science Foundation (No. 2023M731253) and Hubei Provincial Natural Science Foundation (No. 2023AFB487).

References

Hritik Bansal, Karthik Gopalakrishnan, Saket Dingliwal, Sravan Bodapati, Katrin Kirchhoff, and Dan Roth. 2023. Rethinking the role of scale for in-context learning: An interpretability-based case study at 66 billion scale. In *ACL (1)*, pages 11833–11856. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *NeurIPS*.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. A study of language modeling for chinese spelling check. In *SIGHAN@IJCNLP*, pages 79–83. Asian Federation of Natural Language Processing.

Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the fourth ACL-SIGSEM workshop on prepositions*, pages 25–30.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *ACL (Findings)*, pages 4005–4019. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey for in-context learning. *CoRR*, abs/2301.00234.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. Faspell: A fast, adaptable, simple, powerful chinese spell checker based on dae-decoder paradigm. In *W-NUT@EMNLP*, pages 160–169. Association for Computational Linguistics.

Tuo Ji, Hang Yan, and Xipeng Qiu. 2021. Spellbert: A lightweight pretrained model for chinese spelling check. In *EMNLP (1)*, pages 3544–3551. Association for Computational Linguistics.

Ying Jiang, Tong Wang, Tao Lin, Fangjie Wang, Wenting Cheng, Xiaofei Liu, Chenghui Wang, and Weijian Zhang. 2012. A rule based chinese spelling and grammar detection system utility. In *2012 International Conference on System Science and Engineering (ICSSE)*, pages 437–440. IEEE.

Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023a. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*.

Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022. Improving chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity. In *EMNLP*, pages 4275–4286. Association for Computational Linguistics.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023b. On the (in)effectiveness of large language models for chinese text correction. *CoRR*, abs/2307.09007.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and phonologically similar characters in incorrect simplified chinese words. In *COLING (Posters)*, pages 739–747. Chinese Information Processing Society of China.

Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. PLOME: pre-training with misspelled knowledge for chinese spelling correction. In *ACL/IJCNLP (1)*, pages 2991–3000. Association for Computational Linguistics.

Xiaodong Liu, Kevin Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. 2013. A hybrid chinese spelling correction using language model and statistical machine translation with reranking. In *SIGHAN@IJCNLP*, pages 54–58. Asian Federation of Natural Language Processing.

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. 2023. General and domain-adaptive chinese spelling check with error-consistent pretraining. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 22(5):124:1–124:18.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: flexible text editing through tagging and insertion. In *EMNLP (Findings)*, volume EMNLP 2020 of *Findings of ACL*, pages 1244–1255. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. In *EMNLP/IJCNLP (1)*, pages 5053–5064. Association for Computational Linguistics.

Lidia Mangu and Eric Brill. 1997. Automatic rule acquisition for spelling correction. In *ICML*, pages 187–194. Morgan Kaufmann.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinesebert: Chinese pretraining enhanced by glyph and pinyin information. In *ACL/IJCNLP (1)*, pages 2065–2075. Association for Computational Linguistics.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to SIGHAN 2015 bake-off for chinese spelling check. In *SIGHAN@IJCNLP*, pages 32–37. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *ICLR*. OpenReview.net.

Weijian Xie, Peijie Huang, Xinrui Zhang, Kaiduo Hong, Qiang Huang, Bingzhou Chen, and Lei Huang. 2015. Chinese spelling check system based on n-gram model. In *SIGHAN@IJCNLP*, pages 128–136. Association for Computational Linguistics.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, listen, and see: Leveraging multimodal information helps chinese spell checking. In *ACL/IJCNLP (Findings)*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 716–728. Association for Computational Linguistics.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *CIPS-SIGHAN*, pages 220–223. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *ACL*, pages 882–890. Association for Computational Linguistics.

Xiaotian Zhang, Yanjun Zheng, Hang Yan, and Xipeng Qiu. 2023. Investigating glyph-phonetic information for chinese spell checking: What works and what's next? In *ACL (Findings)*, pages 1–13. Association for Computational Linguistics.

A The Detailed Experimental Results

In this section, we introduce the specific experimental results of different in-context learning methods under different semantic information.

Dataset	**Method**	**Detection Level**				**Correction Level**			
		Acc.	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**
SIGHAN15	gpt-3.5-turbo	36.4	22.3	33.0	26.6	34.3	19.3	28.6	23.1
	gpt-3.5-turbo+拼音	45.7	31.2	**46.7**	**37.4**	43.0	27.5	**41.2**	33.0
	gpt-3.5-turbo+部首	48.1	29.7	36.8	32.8	45.6	26.6	31.8	28.4
	gpt-3.5-turbo+结构	45.2	27.8	39.4	32.6	43.6	25.4	36.0	29.8
	gpt-3.5-turbo+笔画	38.7	23.3	35.3	28.1	36.5	20.4	30.9	24.5
	RS-gpt-3.5-turbo	**50.6**	**32.5**	41.6	36.5	**48.1**	**31.2**	40.8	**35.4**
LAW	gpt-3.5-turbo	48.8	30.2	38.8	34.0	46.2	26.2	33.7	29.5
	gpt-3.5-turbo+拼音	64.0	45.4	48.6	47.0	60.8	39.6	42.4	40.9
	gpt-3.5-turbo+部首	**65.0**	45.7	49.4	47.5	**61.4**	39.1	42.4	40.7
	gpt-3.5-turbo+结构	62.4	41.8	43.1	42.5	59.6	36.5	37.7	37.1
	gpt-3.5-turbo+笔画	60.2	40.9	47.8	44.1	55.4	32.9	38.4	35.4
	RS-gpt-3.5-turbo	64.6	**46.6**	**56.1**	**50.9**	60.8	**40.4**	**48.6**	**44.1**
MED	gpt-3.5-turbo	41.4	16.8	30.1	21.5	38.6	13.3	23.9	17.1
	gpt-3.5-turbo+拼音	52.6	**31.0**	37.9	34.1	**48.2**	20.9	32.5	25.5
	gpt-3.5-turbo+部首	50.0	30.3	37.3	33.4	46.2	19.1	29.3	23.1
	gpt-3.5-turbo+结构	46.6	26.0	31.5	28.5	44.1	16.6	26.9	20.5
	gpt-3.5-turbo+笔画	42.4	24.9	30.5	27.5	40.0	16.5	28.1	20.8
	RS-gpt-3.5-turbo	**56.0**	**31.0**	**45.2**	**36.8**	43.3	**25.3**	**39.3**	**30.8**
ODW	gpt-3.5-turbo	63.0	45.8	50.0	47.8	59.2	39.2	42.8	40.9
	gpt-3.5-turbo+拼音	70.4	55.7	58.0	56.8	67.2	49.8	51.9	50.8
	gpt-3.5-turbo+部首	70.8	55.0	60.3	57.6	66.0	46.7	51.2	48.8
	gpt-3.5-turbo+结构	70.8	55.2	58.4	56.8	66.2	46.9	49.6	48.2
	gpt-3.5-turbo+笔画	69.4	53.4	54.6	54.0	65.8	46.6	47.7	47.2
	RS-gpt-3.5-turbo	**72.4**	**59.1**	**64.9**	**61.9**	**70.2**	**50.2**	**57.6**	**53.6**

Table 4: The detailed impact of different rich semantic structures on few-shot learning. RS-gpt-3.5-turbo means RS-LLM on gpt-3.5-turbo. ’拼音’ means phonetic information, ’部首’ means radical information, ’结构’ means structural information, and ’笔画’ means strokes information.

In-context Learning Approaches	**Dataset**	**Method**	**Detection Level**				**Correction Level**			
			Acc.	**Pre.**	**Rec.**	**F1**	**Acc.**	**Pre.**	**Rec.**	**F1**
zero-shot	SIGHAN15	gpt-3.5-turbo	25.9	15.6	26.1	19.6	23.9	13.2	22.0	16.1
		RS-gpt-3.5-turbo	**38.6**	**25.4**	**40.1**	**30.9**	**35.6**	**21.4**	**34.2**	**26.4**
	LAW	gpt-3.5-turbo	32.2	19.9	30.6	24.1	30.2	17.4	26.7	21.0
		RS-gpt-3.5-turbo	**54.8**	**36.7**	**45.1**	**40.5**	**50.4**	**29.7**	**36.5**	**32.8**
	MED	gpt-3.5-turbo	24.6	12.1	22.1	15.7	22.6	9.7	17.7	12.5
		RS-gpt-3.5-turbo	43.0	**25.1**	**41.6**	**31.3**	**40.0**	**21.1**	**35.0**	**26.3**
	ODW	gpt-3.5-turbo	33.8	22.9	34.4	27.5	31.8	20.4	30.5	24.4
		RS-gpt-3.5-turbo	**60.6**	**50.0**	**51.3**	**50.6**	**55.7**	**41.1**	**42.1**	**41.6**
one-shot	SIGHAN15	gpt-3.5-turbo	34.5	19.2	29.0	23.1	32.9	17.1	25.9	20.6
		RS-gpt-3.5-turbo	**46.4**	**30.0**	**42.0**	**35.0**	**43.7**	**26.1**	**36.6**	**30.5**
	LAW	gpt-3.5-turbo	41.2	26.1	37.3	30.7	38.6	22.5	32.2	26.5
		RS-gpt-3.5-turbo	**59.8**	**38.1**	**52.9**	**44.3**	**56.8**	**32.2**	**44.7**	**37.4**
	MED	gpt-3.5-turbo	36.2	14.1	26.1	18.3	23.0	10.3	19.0	13.3
		RS-gpt-3.5-turbo	**51.4**	**30.0**	**44.3**	**35.8**	**47.2**	**23.7**	**35.0**	**28.3**
	ODW	gpt-3.5-turbo	44.9	38.8	46.0	42.1	42.1	28.2	36.0	31.6
		RS-gpt-3.5-turbo	**64.8**	**49.5**	**58.8**	**53.8**	**61.8**	**44.7**	53.1	**48.5**
few-shot	SIGHAN15	gpt-3.5-turbo	36.4	22.3	33.0	26.6	34.3	19.3	28.6	23.1
		RS-gpt-3.5-turbo	**50.6**	**32.5**	41.6	36.5	**48.1**	**31.2**	40.8	**35.4**
	LAW	gpt-3.5-turbo	48.8	30.2	38.8	34.0	46.2	26.2	33.7	29.5
		RS-gpt-3.5-turbo	**64.6**	**46.6**	**56.1**	**50.9**	**60.8**	**40.4**	**48.6**	**44.1**
	MED	gpt-3.5-turbo	41.4	16.8	30.1	21.5	38.6	13.3	23.9	17.1
		RS-gpt-3.5-turbo	**56.0**	**31.0**	**45.2**	**36.8**	**43.3**	**25.3**	**39.3**	**30.8**
	ODW	gpt-3.5-turbo	63.0	45.8	50.0	47.8	59.2	39.2	42.8	40.9
		RS-gpt-3.5-turbo	**72.4**	**59.1**	**64.9**	**61.9**	**70.2**	**50.2**	**57.6**	**53.6**

Table 5: The detailed performance of different number of prompt samples. RS-gpt-3.5-turbo means RS-LLM on gpt-3.5-turbo.