From Spelling to Grammar: A New Framework for Chinese Grammatical Error Correction

Xiuyu Wu[1,2] **Yunfang Wu**[1,3*]
[1]MOE Key Laboratory of Computational Linguistics, Peking University, Beijing, China
[2]School of Software and Microelectronics, Peking University, Beijing, China
[3]School of Computer Science, Peking University, Beijing, China
{xiuyu_wu,wuyf}@pku.edu.cn

Abstract

Chinese Grammatical Error Correction (CGEC) aims to generate a correct sentence from an erroneous sequence, where different kinds of errors are mixed. This paper divides the CGEC task into two steps, namely spelling error correction and grammatical error correction. Specially, we propose a novel zero-shot approach for spelling error correction, which is simple but effective, obtaining a high precision to avoid error accumulation of the pipeline structure. To handle grammatical error correction, we design part-of-speech (POS) features and semantic class features to enhance the neural network model, and propose an auxiliary task to predict the POS sequence of the target sentence. Our proposed framework achieves a 42.11 $F_0.5$ score on CGEC dataset without using any synthetic data or data augmentation methods, which outperforms the previous state-of-the-art by a wide margin of 1.30 points. Moreover, our model produces meaningful POS representations that capture different POS words and convey reasonable POS transition rules.

1 Introduction

Grammatical error correction (GEC) takes erroneous sequences as input and generates correct sentences. In recent years, English GEC task has attracted wide attention from researchers. By employing pre-trained models (Kaneko et al., 2020; Katsumata and Komachi, 2020) or incorporating synthetic data (Grundkiewicz et al., 2019; Lichtarge et al., 2019), the sequence-to-sequence models achieve remarkable performance on English GEC task. Besides, several sequence labeling approaches are proposed to cast text generation as token-level edit prediction (Malmi et al., 2019; Awasthi et al., 2019; Omelianchuk et al., 2020).

Chinese grammatical error correction (CGEC) is less addressed. Previous works adopt ensemble methods by combining seq2seq networks with heuristic rules (Zhou et al., 2018; Fu et al., 2018) or sequence editing approaches (Hinson et al., 2020; Zhang et al., 2022). Different from English, Chinese language utilizes function words instead of affixes to represent forms and tenses, making it hard to design detailed Chinese-specific edit labels. The simple label strategy (for example, *Keep*, *Delete*, *Append_X*) has been proved not competitive with the seq2seq model on CGEC task (Chen et al., 2020; Zhang et al., 2022).

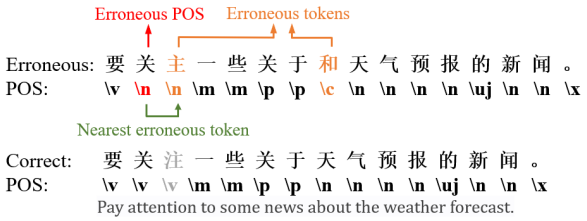

Figure 1: An example of erroneous-correct sentence pair. Black colour: correct tokens. Orange: erroneous tokens. Red: correct tokens with wrong POS tags.

Generally, the errors occurring in Chinese texts can be divided to spelling errors and grammar errors. For example in Figure 1, "主" is a spelling error which should be subtitued to "注". While "和 (and)" relates to grammatical error which should be deteled. According to our statistics on HSK data[1], which is collected from the writing section of Chinese proficiency exam, the proportion of spelling errors is about 18.58%. Unfortunately, most of previous works mix two kinds of errors and adopt the same model to handle them. Moreover, as the basic elements for sentence understanding and processing, spelling errors will influence the usage of high-level features in the CGEC task. Although amounts of linguistic features have been investigated to improve many natural language processing tasks, deep syntactic and semantic knowledge is rarely explored in CGEC.

*Corresponding author.

[1]http://hsk.blcu.edu.cn/Login

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 889 - 902
December 7-11, 2022

Therefore, we propose a new framework for CGEC with two steps: **S**pelling error correction and semantic-enriched **G**rammatical error correction (SG-GEC). We propose a novel zero-shot method for Chinese spelling error correction, by taking advantage of the pre-trained BERT and Chinese phonetic information, which is straightforward but achieves a satisfying precision. Further, we introduce semantic knowledge into the seq2seq model to correct grammatical errors. We carefully analyze the reliability and utility of part-of-speech (POS) in erroneous-correct paired sentences, and design an effective method to integrate POS and semantic representations into the neural network model. Moreover, we introduce an auxiliary task of POS sequence prediction, where a Conditional Random Field (CRF) layer is added to ensure the valid of generated POS sequences and stimulate the model to learn grammar-level corrections.

We conduct extensive experiments on CGEC NLPCC dataset (Zhao et al., 2018). Experimental results show that our proposed zero-shot spelling error correction module achieves a 60.25 precision, which lays a good foundation for further leveraging word-level features. With the pre-trained BART for initialization, our model achieves a new state-of-the-art result of 42.11 $F_{0.5}$ score, which outperforms all previous approaches including pre-trained models and ensemble methods. We also evaluate model performance on CGED-2020 test dataset (Rao et al., 2020) and obtain satisfying results.

To sum, our contributions are as follows:

- We present a new framework for CGEC, which first conducts a preliminary spelling error correction and then performs grammatical error correction with semantic features.
- We propose a novel zero-shot Chinese spelling error correction method, which is straightforward and achieves a high precision.
- We effectively inject semantic knowledge to CGEC at both encoder and decoder, by incorporating POS and semantic class features into the input embeddings, and introducing an auxiliary task of POS sequence generation in the decoding phase.
- Our proposed model obtains a new state-of-the-art result on CGEC task, outperforming previous works by a wide margin without using any data augmentation method.

2 Observation and Intuition

Various types of linguistic features have been exploited in NLP, which bring improvement on different tasks. However, it remains an open issue to introduce linguistic features to GEC. Different from other NLP tasks, the GEC task takes erroneous sentences as input, based on which the extra features might bring noise to the GEC model that harms the performance.

2.1 Part-of-Speech and Grammar Errors

Part-of-speech represents the syntactic function of a word in contexts, which is closely connected with grammar. To bring POS features to the GEC task, the reliability and sensitivity of POS tags to grammar errors should be carefully examined. We conduct such analysis on NLPCC dataset, using Jieba[2] as the POS tagger.

According to our statistics, 88.2% of erroneous sentences have different POS sequences with their paired correct sentences, demonstrating that POS feature is sensitive to grammatical errors. An example is given in Figure 1. We count LCS (Longest Common Sub-sequence) between erroneous-correct sentence pairs. We divide tokens in the erroneous sentence into two types: *Corr-token* which appears in LCS and *Err-token* which does not appear in LCS. Consequently, 98.1% *Corr-tokens* have correct POS tags, proving that the POS tagger could provide precise feature for the correct part of erroneous sentences. As for those 1.9% *Corr-tokens* which have wrong POS tags (red colour in Figure 1), we calculate the average distance between them and the nearest *Err-token* (orange color) and the result is 2.38 tokens. In contrast, the average distance between *Corr-tokens* with correct POS tags and the nearest *Err-token* is 8.59 tokens. This suggests that *Corr-tokens* with wrong POS tags are next to the erroneous part of sentences.

All these statistical results demonstrate that the POS feature is sensitive to the erroneous part meanwhile is robust for the correct part of sentences.

2.2 Semantic Class and Grammar Errors

Word semantic class is a kind of context-free feature, which tells which class each word belongs to according to a semantic dictionary. The dictionary is organized in a tree structure, consisting of different levels of semantic classes. For example, top-3

[2] https://github.com/fxsjy/jieba

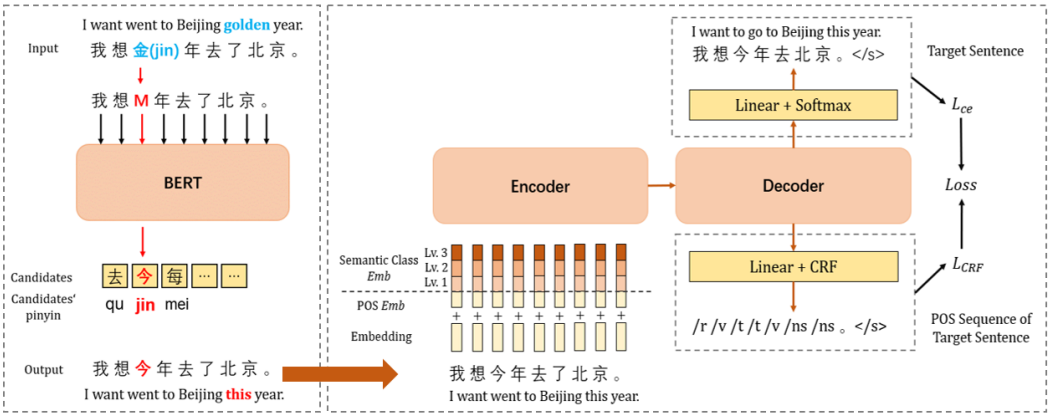

Figure 2: Overview of our new framework for CGEC, which is composed of Spelling Error Correction (left part) and Grammatical Error Correction (right part). **M** refers to the [MASK] symbol in the BERT model.

level semantic class of soup is entity, water and boiled water. By introducing semantic class knowledge, the model could learn the correlation between different semantic classes, and thus correct some semantic collocation errors, such as "冬阴功对外国人的喜爱 (Seafood soup enjoys foreigners.)" . In this example, a kind of food is incorrectly used as the subject which performs the action "enjoy" .

We leverage HIT-CIR Tongyici Cilin (Extended) [3] to provide semantic class knowledge.

3 Zero-shot Spelling Error Correction

We present a new framework for CGEC as shown in Figure 2, which consists of Spelling Error Correction (SEC) and Grammatical Error Correction (GEC). Firstly, we propose a smart zero-shot method for SEC.

Formally, the input erroneous sentence is represented as $X = (x_1, x_2, ...x_n)$, and the target correct sentence is denoted as $Y = (y_1, y_2, ...y_m)$, where n, m mean the length of sentence. We use $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_n)$ to represent the output of the zero-shot spelling error correction module:

$$\tilde{X} = SEC(X) \tag{1}$$

Specially, if a token x_i in X has a relatively high probability of being written incorrectly, it will be substituted with a [MASK] token. Then, the pre-trained BERT model (Devlin et al., 2019) is employed to generate top-3 candidate tokens $V = (v_1, v_2, v_3)$ with high probability. Among the candidates, we select the token that most likely appears in the [MASK] position:

$$\tilde{x}_i = \begin{cases} x_i, & v_j \notin SimSet(x_i) \\ v_j, & v_j \in SimSet(x_i) \end{cases} \tag{2}$$

If there exists more than one token belonging to $SimSet(x_i)$, we choose the token with the highest score generated by BERT. According to the previous study, over 80% spelling errors in Chinese are related to phonological similarity (Liu et al., 2010). So, we set $SimSet(x_i)$ to be the collection of homophones of x_i.

Figure 2 gives an example in the left part. In the given sentence, 金(golden) is suspected to be written incorrectly and substituted with [MASK]. The top three tokens with the highest score generated by BERT are: 去(last), 今(this) and 每(every). Among them, 今/jin shares the same PINYIN with 金/jin. So we replace 金 with 今 in the original sentence.

A problem in SEC is how to decide whether a token x_i is likely to be written incorrectly. Intuitively, the punctuation and commonly used Chinese characters, whose occurrences in the training dataset are over k_c, are less likely to be written incorrectly. We directly keep these tokens unchanged to improve the precision of SEC module and reduce the computational expense. To find out the appropriate threshold value k_c, we conduct experiments on the test set of SIGHAN-2015 (Tseng et al., 2015a), which is designed specially for Chinese spelling error correction and contains 1100 examples collected from Chinese language learners.

As shown in Figure 3, the precision score on SIGHAN test dataset has peaked at 66.1 when $k_c = 80,000$ and gradually declines after $k_c > 120,000$. In order to restrain error accumulation

[3] http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm

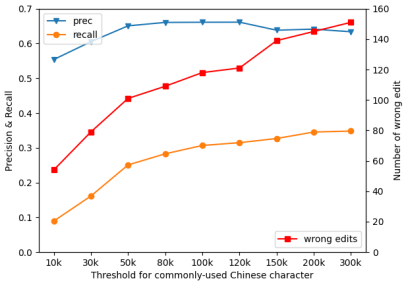

Figure 3: Result of zero-shot spelling correction module evaluated on SIGHAN test dataset.

of the pipeline structure, we hope our SEC module to be high in precision. Accordingly, we set $k_c = 80,000$ in our experiment.

4 Integrating Semantics for Grammatical Error Correction

We adopt the Transformer encoder-decoder architecture for grammatical error correction. To project semantic knowledge to CGEC, at the encoder we incorporate the semantic knowledge representations, and at the decoding we design the POS sequence generation as an auxiliary task.

We add the semantic knowledge embedding $E^{semantics}$ to the original word embedding to serve as the input of encoder:

$$H^{src} = Encoder(E^{word} + E^{semantics}) \quad (3)$$

In the decoding phase, we take the hidden state of timestep *t* to predict the t^{th} token in the target sentence:

$$H_t^{tgt} = Decoder(H_{\leq t-1}^{tgt}, H^{src}) \quad (4)$$

$$P_t^v(w) = softmax(Linear(H_t^{tgt})) \quad (5)$$

where $P_t^v(w)$ is the generation probability of each token.

4.1 Injecting Semantic Features

The semantic knowledge is composed of POS and semantic classes. Please note that, in this stage, the input sequence is $\tilde{X}$, where the spelling error correction has been conducted.

We leverages a POS tagger to obtain the POS tag $\tilde{X}^p = (\tilde{x}_1^p, \tilde{x}_2^p, ...\tilde{x}_n^p)$ for each token in $\tilde{X}$. The embedding of POS tag sequence is computed as:

$$E_i^p = Emb^p(\tilde{x}_i^p), E_i^p \in \mathbb{R}^{d_p} \quad (6)$$

As shown in Table 1, there are different levels of semantic classes to specify a word. We use $\tilde{X}^{c,l} = (\tilde{x}_1^{c,l}, \tilde{x}_2^{c,l}, ...\tilde{x}_n^{c,l})$ to represent the l^{th} level class feature for each token. The high level semantic class brings precise information. If only the high level class feature is extracted, the model will treat words as individual groups and ignore their relations in low levels. Therefore, the semantic representation of token x_i is calculated as the concatenation of embeddings of the first *k-th* levels:

$$E_i^c = E_i^{c,1} \oplus E_i^{c,2} \cdots \oplus E_i^{c,k}, E_i^c \in \mathbb{R}^{d_c \times k} \quad (7)$$

$$E_i^{c,l} = Emb^{c,1}(\tilde{x}_i^{c,l}), E_i^{c,l} \in \mathbb{R}^{d_c} \quad (8)$$

Considering that POS could be regarded as a kind of rough semantic knowledge and be located at the lowest level of semantic class, we concatenate the POS embedding and semantic class embedding to obtain the semantic representation:

$$E_i^{semantics} = E_i^p \oplus E_i^c \quad (9)$$

To make the dimension of semantic embedding equal to that of word embedding d_E, the dimension of POS embedding d_p and that of semantic class embedding d_c are set to:

$$d_p = d_c = \frac{d_E}{k+1} \quad (10)$$

4.2 Predicting POS Sequence

As described in Section 2.1, the wrong POS tags are usually close to the erroneous parts of sentences, which indicates that token-level error correction shares the same target with POS-level error correction. Moreover, POS-level errors are more general, since various types of token-level errors might be mapped to the same on POS-level. Inspired by this observation, we design a sub-task to predict the error-free POS sequence.

At timestep *t*, the generation probability of each token's POS tag is computed utilizing the linear function and softmax:

$$P_t^{pos}(w) = softmax(Linear(H_t^{tgt})) \quad (11)$$

The cross entropy loss is commonly used to stimulate the model to generate a target sequence. However, besides being close to the golden correct POS sequence, the generated POS sequence

itself should be well-formed. To model the dependencies among neighboring POS tags, we adopt Conditional Random Fields (CRF) (Lafferty et al., 2001), under which the likelihood of target POS sequence $Y^p = (y_1^p, y_2^p, ..., y_m^p)$ is computed as:

$$P_{crf}^{pos}(Y^p|X) = \frac{1}{Z(X)} exp\Big(\sum_{t=1}^{m} s(y_t^p) + \sum_{t=2}^{m} t(y_{t-1}^p, y_t^p)\Big) \quad (12)$$

where $s(y_t^p) = P_t^{pos}(w)$, which represents the generation probability of y_t^p.

The value $t(y_{t-1}^p, y_t^p) = M_{y_{t-1}^p, y_t^p}$ denotes the transition score from POS tag y_{t-1}^p to y_t^p, which can be learnt as parameters during the end-to-end training procedure. The Viterbi algorithm (Forney, 1973; Lafferty et al., 2001) is utilized to calculate the normalizing factor $Z(X)$.

4.3 Training Objective

As shown in Figure 2, our model is trained to generate the target sentence and POS sequence simultaneously, and thus the final loss is computed as:

$$Loss = L_{ce} + L_{CRF} \quad (13)$$

$$L_{ce} = -log \sum_{j=1}^{m} P^v(y_j) \quad (14)$$

$$L_{CRF} = -log P_{crf}^{pos}(Y^p|X) \quad (15)$$

5 Experimental Setup

5.1 Dataset and Evaluation Metric

We conduct experiments on the dataset of NLPCC-2018 shared task (Zhao et al., 2018) which contains 1.12 million training samples collected from the language learning platform Lang-8 [4] and 2000 human annotated samples for test. We randomly selected 5,000 instances from training data as the development set. Besides, we changed the format of CGED-2020 test dataset (Rao et al., 2020) to suit our task, and manually corrected 283 word-order errors in CGED-2020 to obtain error-free sentences (Please refer to Appendix A). We evaluate our model on CGED-2020 (1457 samples) as a supplement.

For NLPCC-2018 test dataset, we segment model outputs by the official PKUNLP tool, and adopt the official MaxMatch (M^2) (Dahlmeier and Ng, 2012) scorer to calculate precision, recall and $F_{0.5}$ score. For CGED-2020 test dataset, we apply the simple char-based evaluation using ChERRANT [5] to avoid the influence brought by different word segmentation tools.

5.2 Training Details

Our model is implemented using Fairseq. We average parameters of the last 5 checkpoints. We use BART-base-chinese[6] to initialize our model. We use BERT tokenizer for word tokenization and replace some [unused] tokens with Chinese punctuation. Please refer to Appendix B for more parameter settings.

5.3 Comparing Methods

We compare our model with **YouDao** (Fu et al., 2018), **AliGM** (Zhou et al., 2018) and **BLCU** (Ren et al., 2018) , which are the three top systems in the NLPCC-2018 challenge.

Also, the following previous works are referred as baseline models:

ESD-ESC uses a pipeline structure to firstly detect the erroneous spans and then generate the correct text for annotated spans (Chen et al., 2020).

HRG proposes a heterogeneous approach composed of a LM-based spelling checker, a NMT-base model and a sequence editing model (Hinson et al., 2020).

MaskGEC adds random noises to source sentences dynamically in the training process (Zhao and Wang, 2020).

S2A model combines the output of seq2seq framework and token-level action sequence prediction module (Li et al., 2022).

More recently, (Zhang et al., 2022) enhance the text editing model **GECToR** (Omelianchuk et al., 2020) by using Struct-BERT as its encoder. They also ensemble GECToR with fine-tuned BART model (denotes as **3×Seq2Edit + 3×Seq2Seq**) utilizing edit-wise vote mechanism.

Besides, we finetune **BART** (Lewis et al., 2020) on training dataset and apply MaskGEC as data augmentation method to provide a strong baseline.

[4] https://lang-8.com/

[5] https://github.com/HillZhang1999/MuCGEC/tree/main/scorers/ChERRANT

[6] https://huggingface.co/fnlp/bart-base-chinese

Model	NLPCC-2018			CGED-2020		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
AliGM▲ (Zhou et al., 2018)	41.00	13.75	29.36	-	-	-
YouDao▲ (Fu et al., 2018)	35.24	18.64	29.91	-	-	-
BLCU▲ (Ren et al., 2018)	47.63	12.56	30.57	-	-	-
ESD-ESC (Chen et al., 2020)	37.30	14.50	28.40	-	-	-
S2A model (Li et al., 2022)	36.57	18.25	30.46	-	-	-
HRG▲ (Hinson et al., 2020)	36.79	**27.82**	34.56	-	-	-
MaskGEC (Zhao and Wang, 2020)	44.36	22.18	36.97	-	-	-
Transformer	38.43	12.95	27.58	31.69	11.43	23.39
SG-GEC (Transformer)	44.52	18.28	34.59	32.37	12.04	24.20
BERT-fuse (Kaneko et al., 2020)	42.01	20.24	34.57	31.50	14.99	25.81
GECToR (Zhang et al., 2022)	39.83	23.01	34.75	-	-	-
GECToR (Our implement)	38.76	23.19	34.17	33.33	19.46	29.17
3×Seq2Edit + 3×Seq2Seq▲ (Zhang et al., 2022)	**55.58**	19.78	40.81	-	-	-
BART	46.21	25.14	39.58	38.89	**20.13**	32.78
BART + MaskGEC	48.79	24.03	40.45	40.72	18.63	32.91
SG-GEC (BART init)	50.56	25.24	**42.11**	**40.97**	20.05	**33.90**

Table 1: Performance comparison on the NLPCC-2018 test dataset (Zhao et al., 2018) and CGED-2020 test dataset (Rao et al., 2020). ▲ refers to ensemble model.

6 Results and Analysis

6.1 Overall Performance

Table 1 reports the main evaluation results of our proposed model on NLPCC-2018 and CGED test datasets, comparing with previous researches.

Our proposed SG-GEC model obtains a new state-of-the-art result with a 42.11 $F_{0.5}$ score, which outperforms the previous best single / ensemble model by 5.14 / 1.30 points. Meanwhile, our SG-GEC model surpasses GECToR, which achieves SOTA result on English GEC task. Comparing with the base BART fine-tuned method, our strategy brings a performance gain of 2.53 points. What's more, our SG-GEC model achieves a significant better result in $precision$ among singe models, which is vital for some real-world applications. Without using pre-trained language models, our method outperforms the baseline Transformer by a large margin of 7.01 $F_{0.5}$ points.

Meanwhile, when being initialized by the pre-trained BART, our proposed framework obviously surpasses MaskGEC. It demonstrates that our SG-GEC model brings additional semantic knowledge which is more beneficial to the strong BART model than simple data augmentation methods.

Our model consistently outperforms all other models when evaluated on CGED-2020 test dataset, which proves the generality of our model.

6.2 Ablation Study

We conduct ablation study on NLPCC dataset to evaluate the effect of each module, as shown in Table 2. All the modules bring improvement in model performance. Specially, removing spelling error correction (SEC) results in a sharp decrease of 2.21 $F_{0.5}$ score, for the reason that it not only corrects spelling errors but also offers more reliable POS and semantic class features. Setting embedding of POS / semantic class features to zero leads a decrease of 0.59 / 0.67 $F_{0.5}$ score, which demonstrates that POS and semantic class features bring valid information for grammar error detection. Replacing CRF loss with cross entropy loss leads to a decrease of 0.31 point. Without the subtask of POS generation, the model performance drops from 42.11 to 41.40. It illustrates that predicting POS with CRF layer helps the model to learn grammar-level correction which further improves the performance.

Model	P	R	$F_{0.5}$	Imp.
SG-GEC (BART init)	50.56	25.24	42.11	-
- SEC	49.70	22.30	39.90	- 2.21
- POS *emb*	48.85	25.95	41.52	- 0.59
- Semantic Class *emb*	48.73	25.92	41.44	- 0.67
- POS predict & CRF	49.50	25.02	41.40	- 0.71
- CRF	50.03	25.21	41.80	- 0.31

Table 2: Ablation study of our model on NLPCC dataset. **- CRF** refers to substitute crf loss with cross entropy loss.

6.3 Effect of Spelling Error Correction

In our framework, zero-shot spelling error correction (SEC) is a vital step. We conduct further exper-

Model	*Num.*	P	R	$F_{0.5}$
SEC	192	60.25	5.18	19.27
BART	120	46.21	25.14	39.58
BART + SEC	210	47.70	25.80	40.78
BART + SemF	113	48.11	23.24	39.63
BART + SemF + SEC	210	49.50	25.02	41.40
B-sec	153	25.63	7.63	17.41
BART + B-sec	174	38.98	25.48	35.21
BART + SemF + B-sec	176	39.07	24.33	34.85

Table 3: Effect of spelling error correction. *Num.* refers to the number of corrected spelling error tokens. **SEC** refers to zero-shot spelling error correction module. **B-sec** is a BERT model finetuned on the spelling error correction dataset SIGHAN15 and HybirdSet. + SemF denotes integrating semantic features.

iments to illustrate the effect of this module, and list the results in Table 3.

Our proposed SEC module greatly improves the number of corrected spelling errors, with 90 more tokens over BART and 97 more tokens over the semantic-enriched BART. During the pre-training process of BART model, input tokens are substituted to [MASK] symbols and new tokens are generated without special constraints. Meanwhile, our SEC module intentionally masks misspelled tokens and takes phonetic similarity as constraints when generating new tokens, therefore corrects more spelling errors and achieves high precision score.

If semantic feature embeddings are directly added on BART without utilizing the SEC module, the number of corrected spelling errors will drop from 120 to 113. Because spelling errors influence word segmentation and thus lead to erroneous POS and semantic class features at the position of misspelled tokens. In contrast, our high precision SEC lays a solid foundation for the further semantic information injection. After applying the SEC module, BART + SEC + SemF (semantic features) obtains larger improvement in model performance.

We also compare our zero-shot SEC module with a BERT model finetuned on the spelling error correction dataset SIGHAN-2015 (Tseng et al., 2015a). Our SEC module strictly focuses on correcting spelling errors and achieves a high precision. However, beside spelling errors, the finetuned BERT model automatically corrects other types of errors, leading to a high recall but low precision score. As the first step of pipeline structure, the low precision brings huge noise to the subsequent module and thus damages the final performance.

6.4 Analysis on POS representations

The part-of-speech feature is closely connected with grammar. In our model, we inject POS embedding in encoder and predict the correct POS sequence in decoder, which enable our model to learn a better POS representation.

To investigate the POS representations, we calculate the nearest neighbours to each POS tag by computing the cosine distance between embedding vectors, and list the results in Table 4. For each POS tag, most of their nearest tokens have the corresponding part-of-speech. It demonstrates that our POS embedding could capture general features of tokens sharing the same part-of-speech, which benefits our model and shows potential for other NLP applications.

POS Tag	Top-3 nearest tokens		
noun	栈 storehouse	障 obstacle	浆 liquid
verb	想 think	离 leave	做 do
adjective	脆 crisp	傻 foolish	幸 lucky
adverb	再 again	也 also	永 always
pronoun	我 I	他 he	飞 fly
preposition	对 for/towards	把 *prep.*	为 for

Table 4: Top-3 nearest tokens to POS tags.

In our model, CRF is essential to capture neighboring POS dependencies of target sequences. We visualize the POS transition matrix that the CRF layer has learnt in Figure 4. Interestingly, several grammar rules could be found. For example, $preposition$ is usually followed by $noun$, $pronoun$ or $spacename$, but it has a low probability of transiting to punctuation (the end of a sentence). $Adjective$ usually occurs before $noun$ but seldom connects with $preposition$. This POS knowledge enables our model to generate grammatical sentences.

6.5 Case Study

Table 5 provides an example output of our SGGEC model comparing with BART. Our SEC module firstly corrects the spelling error in this sentence. Benefiting by this, seq2seq model corrects the grammatical error subsequently. However, the BART model fails to correct both spelling and grammatical errors in this sentences. More cases are listed in Appendix F.

We list two cases in Table 6 to show the effect of our semantic class feature. When object and verb are mismatched (Case 1) or verb is missing (Csed

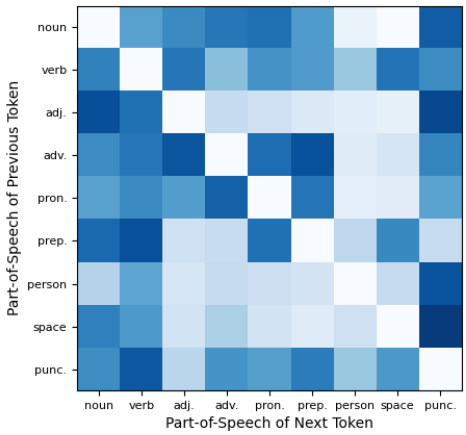

Figure 4: POS transition matrix in the CRF layer. Darker colour refers to higher transition probability.

Type	Sample
SRC	由于羊驼毛的价格比羊羔毛低廉且更具优的保暖性...
TGT	由于羊驼毛的价格比羊羔毛低廉且更具有(delete-的)保暖性...
BART	由于羊驼毛的价格比羊羔毛低廉且更具优的保暖性...
SEC	由于羊驼毛的价格比羊羔毛低廉且更具有的保暖性...
SG-GEC	由于羊驼毛的价格比羊羔毛低廉且更具有(delete-的)保暖性...
Translation	Because paco wool is cheaper and warmer than lamb wool...

Table 5: Case study of our model. Red / blue color refers to correction of spelling / grammatical error.

2), our model could correct these errors benefited from information provided by semantic class feature. When meeting rarely used words, for example 罪魁祸首(chief culprit), semantic class feature might provide extra information learning from examples which contain 主犯(principal criminal) or 要犯(important crimial) and help model to replace verb 了解(know about) with 了结(kill).

7 Related Work

7.1 Grammatical Error Correction

Seq2seq generation model and edit label prediction model are two mainstream models for GEC task. Benefiting by the rapid gains in hardware and high quality dataset, Transformer-based seq2seq models (Junczys-Dowmunt et al., 2018; Katsumata and Komachi, 2020; Kaneko et al., 2020) outperform traditional CNN and RNN-based model structures (Xie et al., 2016; Yuan and Briscoe, 2016; Chollampatt and Ng, 2018). Copy mechanism and subtask is also introduced to seq2seq model (Zhao et al., 2019). LaserTagger (Malmi et al., 2019) treats the GEC task as text edit task and predicts *Keep*, *Delete* and *Append_#* for each token in erroneous sentences to represent different edit operation. PIE (Awasthi et al., 2019) and GECToR (Omelianchuk et al., 2020) manually design detailed English-specific labels, regarding case and tense. Synthetic data is generated to enhance model performance (Ge et al., 2018; Grundkiewicz et al., 2019; Lichtarge et al., 2019). Besides two mainstream model structure, ESD-ESC (Chen et al., 2020) firstly detects erroneous spans and generates correct contents only for annotated spans. TtT model (Li and Shi, 2021) directly predicts each tokens in correct sentences given erroneous sentence.

CGEC task is less addressed. Release of NLPCC-2018 dataset (Zhao et al., 2018) attracts much attention from participated teams, where top 3 systems are AliGM (Zhou et al., 2018), YouDao (Fu et al., 2018) and BLCU (Ren et al., 2018). HRG combines spelling checker, NMT-base model and sequence editing model (Hinson et al., 2020). However, spelling checker in HRG is based on language model which could not make full use of context. Zhao and Wang proposed data augmentation method MaskGEC, which adds random noise to input sentence dynamically in training process. S2A model combines seq2seq and sequence editing model by combining prediction probability of words and edit labels (Li et al., 2022). Zhang et al. ensembles seq2seq model and sequence editing model by edit-wise vote mechanism and achieves the state-of-the-art on NLPCC-2018 dataset.

7.2 Chinese Spelling Error Correction

Chinese spelling error correction is firstly tackled with CRF or HMM models (Tseng et al., 2015b; Zhang et al., 2015). In recent neural network models, phonological and graphic knowledge is introduced to help detecting and correcting Chinese spelling errors (Hong et al., 2019; Huang et al., 2021; Cheng et al., 2020). The pre-trained BERT model is also utilized to generate candidate sentences (Hong et al., 2019; Zhang et al., 2020).

Different from these models, we locate possibly misspelled tokens based on rule instead of neural network. We directly choose the homophone of masked token from candidates generated by BERT. Knowledge is utilized more explicitly in our module. More importantly, our method is zero-shot, without using any labeled data.

Type	Sample
keyword	罪魁祸首-chief culprit
group words	主犯-principal criminal 要犯-important criminal 正凶-principal murderer
SRC	我就在此了解(know about)了你这罪魁祸手(chief culprit)，平平风气！
TGT	我就在此了结(kill)了你这罪魁祸首(chief culprit)，平平风气！
BART	我就在此了解(know about)了你这罪魁祸手(chief culprit)，平平风气！
SG-GEC	我就在此了结(kill)了你这罪魁祸首(chief culprit)，平平风气！
Translation	I'm going to kill you chief culprit right here to improve social climate.
keyword	年级-grade
group words	班级-class 高年级-senior year
SRC	时间过的很快，我已经变成(become)了三年级(the third grade)。
TGT	时间过得很快，我已经变成(become)了三年级的学生(third-grade student)。
BART	时间过的很快，我已经变成(become)了三年级(the third grade)。
SG-GEC	时间过的很快，我已经上(is in)三年级(the third grade)了。
Translation	Time passed quickly, I have become a third-grade student.
keyword	意思-meaning
group words	意义-significance 含义-implication
SRC	背完生词之后，再读课文，那么更容易生词的用法和意思(meaning)。
TGT	背完生词之后，再读课文，那么更容易记住(remember)生词的用法和意思(meaning)。
BART	背完生词之后，再读课文，那么生词的用法和意思(meaning)就更容易了。
SG-GEC	背完生词之后，再读课文，会更容易理解(understand)生词的用法和意思(meaning)。
Translation	Reading the passage after reciting the new words makes it easier to remember the usage and the meaning of the new words。

Table 6: Case study of our model. Group words refer to words which share the same semantic class with the keyword. Blue / red color refers to verb / object. Green color refers to modification of verb or object.

8 Conclusion

In this paper, we divide CGEC into two consecutive tasks: spelling error correction and grammatical error correction. We propose a zero-shot spelling error correction method, utilizing the pre-trained BERT model and taking advantage of Chinese phonological knowledge. It achieves a high precision score to avoid error accumulation in the pipeline structure. Based on the careful analysis on real data, we inject proper semantic features into the encoder. And at the same time, we generate correct POS sequence as a sub-task to help generate correct sentences, where CRF is applied to guarantee the validness of the generated POS sequence. Initialized by the pre-trained BART model, our proposed framework achieves a new state-of-the-art result on CGEC task, outperforming the previous best result by a large margin.

Limitation

Our zero-shot spelling error correction module is specifically designed for Chinese language. Meanwhile, the POS tagger and vocabulary of semantic class we used in SG-GEC model cannot be directly applied to other languages. To some degree, it makes SG-GEC model as a language-specific model. We try to find matched resources in English language and conduct experiments on English GEC dataset. The result is reported in Appendix E. It demonstrates that introducing semantic features after spelling check and employing sub-task of POS correction with CRF layer, which is the main idea of our work, could benefit GEC task of other languages.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (62076008), the Key Project of Natural Science Foundation of China (61936012) and the National Hi-Tech RD Program of China (No.2020AAA0106600).

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *EMNLP/IJCNLP (1)*, pages 4259–4269. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *BEA@ACL*.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span

detection and correction. In *EMNLP (1)*, pages 7162–7169. Association for Computational Linguistics.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check. In *ACL*, pages 871–881. Association for Computational Linguistics.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *AAAI*, pages 5755–5762. AAAI Press.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *HLT-NAACL*, pages 568–572. The Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *BEA@NAACL-HLT*.

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.

G. D. Forney. 1973. The Viterbi algorithm. *Proceedings of IEEE*, 61(3):268–278.

Kai Fu, Jin Huang, and Yitao Duan. 2018. Youdao's winning solution to the nlpcc-2018 task 2 challenge: A neural machine translation approach to chinese grammatical error correction. In *NLPCC (1)*, volume 11108 of *Lecture Notes in Computer Science*, pages 341–350. Springer.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *ACL (1)*, pages 1055–1065. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *BEA@ACL*, pages 252–263. Association for Computational Linguistics.

Charles Hinson, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Heterogeneous recycle generation for chinese grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2191–2201. International Committee on Computational Linguistics.

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.

Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao. 2021. Phmospell: Phonological and morphological knowledge guided chinese spelling check. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5958–5967.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. In *NAACL-HLT*, pages 595–606. Association for Computational Linguistics.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *ACL*, pages 4248–4254. Association for Computational Linguistics.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *CoRR*, abs/2005.11849.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics.

Jiquan Li, Guo Junliang, Yongxin Zhu, Xin Sheng, Deqiang Jiang, Bo Ren, and Linli Xu. 2022. Sequence-to-action: Grammatical error correction with action guided sequence generation. In *AAAI*. AAAI Press.

Piji Li and Shuming Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for chinese grammatical error correction. In *ACL/IJCNLP*.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *NAACL-HLT (1)*, pages 3291–3301. Association for Computational Linguistics.

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.

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.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *IJCNLP*, pages 147–155. The Association for Computer Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem N. Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector - grammatical error correction: Tag, not rewrite. In *BEA@ACL*, pages 163–170. Association for Computational Linguistics.

Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.

Hongkai Ren, Liner Yang, and Endong Xun. 2018. A sequence to sequence learning for chinese grammatical error correction. In *NLPCC (2)*, volume 11109 of *Lecture Notes in Computer Science*, pages 401–410. Springer.

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

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

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. Neural language correction with character-based attention. *CoRR*, abs/1603.09727.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *ACL*.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *HLT-NAACL*, pages 380–386. The 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.

Shuiyuan Zhang, Jinhua Xiong, Jianpeng Hou, Qiao Zhang, and Xueqi Cheng. 2015. Hanspeller++: A unified framework for chinese spelling correction. In *SIGHAN@IJCNLP*, pages 38–45. Association for Computational Linguistics.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a multi-reference multi-source evaluation dataset for chinese grammatical error correction. In *Proceedings of NAACL-HLT*, Online. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *NAACL-HLT (1)*, pages 156–165. Association for Computational Linguistics.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 439–445. Springer.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *AAAI*, pages 1226–1233. AAAI Press.

Junpei Zhou, Chen Li, Hengyou Liu, Zuyi Bao, Guangwei Xu, and Linlin Li. 2018. Chinese grammatical error correction using statistical and neural models. In *NLPCC (2)*, volume 11109 of *Lecture Notes in Computer Science*, pages 117–128. Springer.

A Annotation of CGED-2020 Dataset

There exist 283 word-order errors in CGED-2020 test dataset. We first correct other three types of errors in the sentences according to the golden answer and mark the start and end points of word-order errors. Two annotators are asked to correct the error by adjusting the order of the tokens between start and end point. For 87% of erroneous sentences, correction results of two annotators are consistent with each other. For the other 13% sentences, a third annotator is asked to select the better one from two different corrected sentences.

B Hyperparameters

The detailed hyperparameter settings are listed in Table 7.

C Effect of Semantic Features

We investigate the effect of each single feature as well as different approaches for feature combination, and the results are shown in Table 8. Both

Hyper-parameters	Value
pretrained model	bart-base-chinese
dropout	0.1
learning rate	3e-5
optimizer	Adam(β_1=0.9, β_2=0.999, ϵ=1e-8)
lr scheduler	polynomial decay
warmup updates	500
total number updates	20000
max tokens	4096
update-freq	2
max epochs	10
loss function	cross entropy
beam size	12
Num. of parameters	117 millions
device	two NVIDIA RTX 2080 GPUs
runtime	4.5 hours

Table 7: Hyper-parameter settings in our model.

POS and semantic class bring helpful knowledge for GEC task. The Level-3 semantic class feature outperforms other single features, which provides more detailed classification information of semantic knowledge. Compared with accumulating, concatenating all features obtains a slightly better result.

Model	Num.	P	R	$F_{0.5}$	Imp.
BART + SEC	-	47.70	25.80	40.78	
+ *POS*	44	49.17	25.07	41.24	+ 0.46
+ *Class Lv.1*	18	48.78	25.44	41.22	+ 0.44
+ *Class Lv.2*	101	48.91	25.50	41.32	+ 0.54
+ *Class Lv.3*	1431	49.50	24.91	41.34	+ 0.56
+ *accum. All*	-	49.01	25.50	41.38	+ 0.60
+ *concat. All*	-	49.50	25.02	41.40	+ 0.62

Table 8: Effect of different semantic features. *Class Lv.k* refers to the *k-th* level semantic class. *accum. All* / *concat. All* denotes the final semantic representation is obtained by accumulating / concatenating all features' embeddings. **Num.** refers to number of different values of a specific feature.

D Effect of Semantic Sequence Prediction

To evaluate the effect of our proposed multitask learning framework, we exploit different semantic sequences as targets of generation, and the results are shown in Table 9. Employing POS sequence generation as a sub-task outperforms other auxiliary tasks of semantic class sequence generation, which brings a further performance gain over the strong model of BART+SEC+SemF. POS sequence is more generalized and convey much more syntactic information. By learning to generate correct POS sequence, model could learn grammar-level

Model	P	R	$F_{0.5}$	Imp.
BART+SEC+SemF	49.50	25.02	41.40	
+ *POS pred*	50.56	25.24	42.11	+ 0.71
w/o CRF	50.03	25.21	41.80	+ 0.40
+ *Class Lv.1 pred*	49.57	24.70	41.41	+ 0.01
w/o CRF	49.57	24.70	41.26	- 0.14
+ *Class Lv.2 pred*	49.20	24.85	41.19	- 0.21
w/o CRF	50.29	23.11	40.71	- 0.69

Table 9: Effect of different types of sequence generation as a sub-task. *POS pred* / *Class Lv.k pred* refers to employ a sub-task to predict POS / *k-th* level semantic class sequence. *w/o CRF* represents the standard cross entropy loss is applied without using CRF.

correction instead of token-level correction. Predicting semantic class sequence does no benefit the performance of model. Firstly, semantic class is a context-free feature based on words, which means predicting sequence semantic class roughly equals to predicting sequence of token. What's more, there are about 15% words in target sentences having no semantic class in the lexicon, which will influence the training process.

E Experiment on English GEC dataset

For English GEC task, following Bryant et al. (2019), we use Lang-8 Corpus of Learner English (Mizumoto et al., 2011), FCE (Yannakoudakis et al., 2011), NUCLE (Dahlmeier et al., 2013) and W&I+LOCNESS (Bryant et al., 2019) as training data, CoNLL-2013 test set as dev set and evaluate on CoNLL-2014 (Ng et al., 2014) test set.

In Chinese language, token might be incorrectly written as its homophone. Meanwhile, in English language, spelling mistakes usually caused by missing or mis-writing letters. Our phonological knowledge based zero shot-spelling error correction module could not be directly applied to English language. Spelling errors in English language cause out-of-vocabulary words, which makes it easier to be detected and corrected compared with Chinese. Therefore, we simply utilize a spelling checker[7] based on dictionary and edit distance to substitue zero-shot SEC module in SG-GEC.

We use NLTK[8] as POS tagger. For semantic class knowledge, we could not find exactly matched resources in English language. We design two alternative solutions:

[7] https://github.com/barrust/pyspellchecker

[8] https://www.nltk.org/

- **zero-class-feature** We set the embedding of semantic class features to zero during both training and inference process.
- **Wordnet-class-feature** We use WordNet[9] to get the semantic class features of a word by recursively searching the hypernym of this word. Number of values in *1rt* / *2nd* / *3rd* level semantic class is 148 / 685 / 9852.

We use BART-base[10] to initialize our model.

Model	P	R	$F_{0.5}$
spelling checker	56.07	2.94	12.14
BART	70.10	40.16	61.00
BART + spelling checker	69.72	41.27	61.27
SG-GEC (BART init)			
zero-class-feature	68.80	43.96	61.82
Wordnet-class-feature	69.01	44.07	62.00

Table 10: Experiment on English GEC dataset.

Table 10 demonstrates that spell checker brings little benefit to BART-finetuned model on English GEC task. One reason is that spelling error in English causes out-of-vocabulary words, which is easily to detect. As shown in Table 11, misspelled out-of-vocabulary words are usually divided into BPE level in BART model.

Correct	Misspelled	Tokenized
potential	potetial	pot ##et ##ial
responsibility	resposiblity	resp ##os ##ibl ##ity
hundreds	hundrends	h ##und ##rend ##s

Table 11: Examples of misspelled words in English GEC dataset.

By introducing POS feature and sub-task of POS correction with CRF layer while setting embedding of semantic class features to zero, our model achieves 61.82 $F_{0.5}$ score, which outperforms BART model. Semantic class features provided by Wordnet also slightly improve the performance of the model. Wordnet focuses on modeling relations between words instead of classification of words. The same level semantic class feature of two different words might be different in scale. For example, root hypernym of "people" is "entity.n.01" while root hypernym of "get" is "get.v.01", which might brings influence to the model.

[9] https://wordnet.princeton.edu/

[10] https://huggingface.co/facebook/bart-base

Experimental result on English GEC dataset demonstrates that our proposed SG-GEC model could also benefit GEC task of other languages.

F More Case Studies

We list five more cases in Table 12 to demonstrate effectiveness of our pipeline structure. BART model might easily miss grammatical error (Case 1, Case 2) or spelling error (Case 3, Case 4) because of mixing spelling error and grammatical error correction together. It might be misguided by erroneous token (Case 5).

Type	Sample
SRC	我认为空气污染是跟我们的生活密切的问题，所以一定要最优先解决，优其是像北京那样的大城市。
TGT	我认为空气污染是跟我们的生活密切相关的问题，所以一定要最优先解决，尤其是像北京那样的大城市。
BART	我认为空气污染是跟我们的生活密切的问题，所以一定要最优先解决，尤其是像北京那样的大城市。
SEC	我认为空气污染是跟我们的生活密切的问题，所以一定要最优先解决，尤其是像北京那样的大城市。
SG-GEC	我认为空气污染是跟我们的生活密切相关的问题，所以一定要最优先解决，尤其是像北京那样的大城市。
Translation	I think air pollution is the problem that are closely related to our lives. Therefor it should be solved as a matter of top priority, especially for metropolis like Beijing.
SRC	人为了生存不管是干静的空气，污染的空气都要呼吸。
TGT	人为了生存不管是干净的空气，还是污染的空气都要呼吸。
BART	人为了生存不管是干静的空气，还是污染的空气都要呼吸。
SEC	人为了生存不管是干净的空气，污染的空气都要呼吸。
SG-GEC	人为了生存不管是干净的空气，还是污染的空气都要呼吸。
Translation	In order to survive, human need to breathe air no matter it is fresh or polluted.
SRC	学校里的草场上还有一点的人来运动。
TGT	学校里的操场上还有一些人在运动。
BART	学校里的草场上还有一些人来运动。
SEC	学校里的操场上还有一点的人来运动。
SG-GEC	学校里的操场上还有一些人来运动。
Translation	In the school playground, there are some people coming for doing exercise.
SRC	几个仙女来蟠桃圆摘桃时，告诉了孙悟空王母要做蟠桃盛会。
TGT	几个仙女来蟠桃园摘桃时，告诉了孙悟空王母要办蟠桃盛会。
BART	几个仙女来蟠桃圆摘桃时，告诉了孙悟空王母要举办蟠桃盛会。
SEC	几个仙女来蟠桃园摘桃时，告诉了孙悟空王母要做蟠桃盛会。
SG-GEC	几个仙女来蟠桃园摘桃时，告诉了孙悟空王母要举办蟠桃盛会。
Translation	When fairies coming to peach orchard to pick peaches, they told Monkey King that the Queen Mother would hold the Peach Banquet.
SRC	这些地方征明中国灿烂的文化和历史。
TGT	这些地方证明了中国灿烂的文化和历史。
BART	这些地方象征着中国灿烂的文化和历史。
SEC	这些地方证明中国灿烂的文化和历史。
SG-GEC	这些地方证明了中国灿烂的文化和历史。
Translation	These places prove that China have splendid culture and history.

Table 12: Case study of our model. Red / blue color refers to correction of spelling / grammatical error.