applied sciences

MDPI

Article

Visual and Phonological Feature Enhanced Siamese BERT for Chinese Spelling Error Correction

Yujia Liu [1], Hongliang Guo [2], Shuai Wang [3] and Tiejun Wang [1,*]

[1] School of Chemical Engineering and Light Industry, Guangdong University of Technology, Guangzhou 510006, China; yujia.liu@gdut.edu.cn
[2] School of Automation Engineering, University of Electronic Science and Technology of China, Xiyuan West Road 2006, Chengdu 611731, China; guohl1983@uestc.edu.cn
[3] Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China; wangshuai@smail.nju.edu.cn
* Correspondence: tjwang@gdut.edu.cn

Abstract: Chinese Spelling Check (CSC) aims to detect and correct spelling errors in Chinese. Most CSC models rely on human-defined confusion sets to narrow the search space, failing to resolve errors outside the confusion set. However, most spelling errors in current benchmark datasets are character pairs in similar pronunciations. Errors in similar shapes and errors which are visually and phonologically irrelevant are not considered. Furthermore, widely-used automatically generated training data in CSC tasks leads to label leakage and unfair comparison between different methods. In this work, we propose a feature (visual and phonological) enhanced siamese BERT to (1) correct spelling errors without using confusion sets; (2) integrate phonological and visual features for CSC by a glyph graph; (3) improve performance for unseen spelling errors. To evaluate CSC methods fairly and comprehensively, we build a large-scale CSC dataset in which the number of samples in different error types is the same. The experimental results show that the proposed approach achieves better performance compared with previous state-of-the-art methods on three benchmark datasets and the new error-type balanced dataset.

Keywords: natural language processing; Chinese spelling check; graph; BERT

Citation: Liu, Y.; Guo, H.; Wang, S.; Wang, T. Visual and Phonological Feature Enhanced Siamese BERT for Chinese Spelling Error Correction. *Appl. Sci.* **2022**, *12*, 4578. https://doi.org/10.3390/app12094578

Academic Editor: Valentino Santucci

Received: 6 March 2022
Accepted: 20 April 2022
Published: 30 April 2022

1. Introduction

Chinese spelling errors are common in daily life, due to the similarity between characters. In Chinese, there are many characters similar in phonology and visual shape, but different in semantics. These spelling errors are typically caused by careless human writing, automatic speech recognition, or optical character recognition systems. Besides, misspellings are intentionally used by perpetrators to avoid automatic detection in current social platforms, such as spam broadcasting, malicious advertising, etc. Therefore, detecting and correcting such misuse of the Chinese language are important tasks in real-world applications. An effective method for CSC is not only useful in natural language tasks such as speech recognition, word recognition, and grammatical error correction but also has potential value in anti-spam problems such as fraud detection, advertisement detection, etc.

Unlike English, Chinese spelling error correction is a challenging task due to the characteristics of Chinese characters. Chinese texts consist of many pictographic characters without word delimiters. The semantic meaning of each character changes dramatically when the context changes. Besides, the pronunciation of a character depends on the context. In this sense, a CSC model is required to both understand the semantics and integrate the surrounding information (i.e., pronunciation, character structure).

Traditional methods have been employed for CSC. For example, previous works [1–3] employed the traditional machine learning [4,5] and other deep learning models [6]. Also, sequence-to-sequence models [7] have been proposed for spelling error correction by transforming an input sentence into a new sentence with spelling errors corrected. Recently, several methods have been introduced to exploit external information of character similarity, which rely on a human-defined confusion set [7,8]. The confusion set is a dictionary containing similar character pairs. In ACL 2019, confusion set-guided Pointer Networks [7] used a pointer network to copy similar characters from the confusion set. In ACL 2020 [8] a spelling check convolutional graph network was presented by constructing similarity graphs using the confusion set. These methods attempt to model the relationship between characters based on the confusion set. However, the similarity information extracted from the confusion set is limited. The confusion set only provides information about whether two characters are similar, but does not present phonological and visual features of Chinese characters. Furthermore, a human-defined confusion set only covers a small subset of Chinese characters due to the human annotation, which can hardly correct the misspellings of unknown characters.

Addressing the problems of human annotation and unknown characters, we propose Feature Enhanced BERT (FE-BERT) to integrate phonological and visual similarity for CSC without using a confusion set. Specifically, we construct a glyph graph over Chinese characters for capturing the shape similarity between characters, by leveraging the intrinsic component structure of Chinese characters. Instead of similar character pairs set by human annotation to capture the shape similarity, our glyph graph requires no human annotation. Furthermore, the glyph graph utilizes the decomposition structure of Chinese characters, which can cover more characters compared to the confusion set. The glyph graph is pre-trained to generate a vector representation for each character as its shape feature. Combining the shape and pronunciation features with BERT [9], FE-BERT can adequately leverage the similarity knowledge and generate the right corrections accordingly. Considering spelling errors which are visually and phonologically irrelevant, we adopt a siamese structure to combine a vanilla BERT and a FE-BERT, namely FES-BERT. As depicted in Table 1, FES-BERT can completely correct the spelling errors led by both the pronunciation similarity and the shape similarity.

Table 1. A CSC data sample from SIGHAN 2015 [10] with ID A2-1515-1, the incorrect/correct characters are in orange/blue. A BERT model modifies the text into a sentence that is semantically reasonable but dissimilar in pronunciation. By incorporating both phonological and visual similarities, our new method FES-BERT can generate a sentence that is both semantically sensible and phonically similar to the original sentence. The sentence output from FES-BERT means "Can you tell me, does this store have Chinese books?", while the sentence Output from BERT means "Can you tell me, does this store have a Chinese department?".

Input (phonics)	你可以告诉我那家书店有中文数马? nǐ kě yǐ gào sù wǒ nà jiā shū diàn yǒu zhōng wén shù mā
BERT (phonics)	你可以告诉我那家书店有中文系吗? nǐ kě yǐ gào sù wǒ nà jiā shū diàn yǒu zhōng wén shū mā
FES-BERT (phonics)	你可以告诉我那家书店有中文书吗? nǐ kě yǐ gào sù wǒ nà jiā shū diàn yǒu zhōng wén shū mā

Experimental results on benchmark datasets and our error type balanced CSC dataset show that our model outperforms BERT and previous state-of-the-art models [7–9,11,12]. In summary, our contributions are as follows:

- We construct a novel glyph graph to model visual similarity between Chinese characters by leveraging the intrinsic component structure of Chinese characters. This method can cover almost all Chinese characters without the need for a confusion set or human annotation.
- For fair and comprehensive performance evaluation, we build a new CSC dataset in which the amount of samples in different error types is the same. Half of the errors in the test set are included by the confusion set used by [8] while the others are outside

the confusion set. Besides, to evaluate performance on new errors, half of the errors in the test set are not included in the training set.
- We incorporate phonological and visual features for CSC and reach a balance between Chinese characters' external features and semantic features via the proposed FES-BERT. Experimental results show that our model achieves better performance compared to previous SOTA models.

2. Related Work

2.1. Unsupervised Methods

The CSC task attracts much attention from the community in recent years. Earlier works in CSC mainly focused on unsupervised methods [1,13]. Character-level n-gram language models were used to detect potential misspelled characters with low probabilities below some pre-defined threshold. To be specific, [1] performs a four-stages procedure to handle spelling check, including character-level n-gram language model, pronunciation and shape-based candidate set generation, candidate corrections filter, and find the best candidate. In [13], the spelling correction process includes two cascaded components: spelling error detection and spelling error correction. In spelling error detection phase, characters with a low score under the language model are regarded as potential error characters. Independent characters after automatic word segmentation are also collected as potential error characters. In the spelling error correction phase, a candidate set will be generated for each error character. The candidate set generation is based on similar pronunciation or shape dictionary (confusion set). The generated candidate set will be filtered by testing if the candidate character can construct a legal word with neighbor characters. After filtering, the character with the highest score calculated by the language model will be selected as the best candidate. However, such unsupervised methods cannot model the semantic context, which is important information in CSC tasks.

2.2. Supervised Methods

Several supervised methods have been proposed to tackle this problem. They regard the CSC task as a sequence labeling problem [14–16], in which RNN based models are adopted. Confusion set guided Pointer Networks [7] used a pointer network to copy similar characters from the confusion set directly. A chunk-based framework was proposed in [17] to correct single-character and multi-character word errors uniformly.

Recently, BERT[9], the masked language representation model, has been successfully applied in CSC task. FASPell [11] fine-tunes a pre-trained BERT to produce character candidates and then use a confidence-similarity decoder to filter the candidates. Candidate characters with high contextual confidence and similarity from the original character are more likely to be the best. Confusionset-guided Pointer Networks [7] proposed a Seq2Seq model to copy an input sentence to a corrected new sentence through a pointer network. Soft-Masked BERT [12] consists of a network for error detection based on GRU [18] and a network for error correction based on BERT, with the former being connected to the latter with a soft-masking technique. The character embeddings of potential error characters are masked via a soft pointer to help BERT select the best candidate.

Graph-based methods have also been proposed to utilize external information of characters for CSC task. SpellGCN[8] with graph convolutional network (GCN) aims to capture similarity information between characters and it is able to achieve the SOTA results on benchmarks. Specifically, two graphs are built for pronunciation and shape similarities correspondingly. An attentive graph is used to combine node representations of the two graphs. For characters in a given input text, SpellGCN uses BERT to extract context representations. Then, similarities between these context representations and node representations extracted by GCN are measured to find the correct answer. SpellGCN relies on the confusion set, which requires human annotation and only covers a subset of Chinese characters.

3. Approach

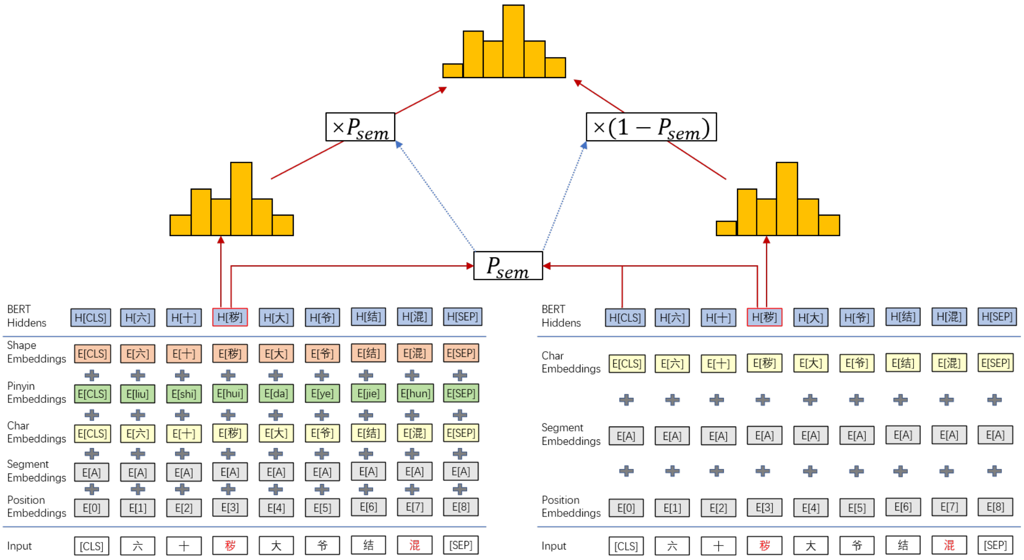
In this section, we describe the CSC task and introduce the Feature-enhanced Siamese BERT (FES-BERT) which is able to jointly learn contextualized representations of characters' shape, pronunciation, and semantic features.

3.1. Problem Formulation

Chinese Spelling Check can be formalized as a generation problem by modeling the probability $P(Y|X)$. Given a text sequence $X = \{x_1, x_2, \ldots, x_l\}$ of l characters containing misspelled characters, the goal of our task is to transform the input sentence into a correct text sequence $Y = \{y_1, y_2, \ldots, y_l\}$ in which the wrong characters are recognized and corrected.

3.2. Model Overview

The framework of the proposed method is depicted in Figure 1. It consists of two components, i.e., a Feature-enhanced BERT (FE-BERT) for visual and phonological errors and a vanilla BERT for visually and phonologically irrelevant typos. The prediction results of the BERT and the FE-BERT are summed up through a dynamic soft pointer to reach a balance between semantic constraints and external feature constraints. The details are shown in Algorithm 1.



$\times P_{sem}$
$\times(1 - P_{sem})$
P_{sem}
BERT Hiddens
H[CLS] H[六] H[十] H[秽] H[大] H[爷] H[结] H[混] H[SEP]
Shape Embeddings
E[CLS] E[六] E[十] E[秽] E[大] E[爷] E[结] E[混] E[SEP]
Pinyin Embeddings
E[CLS] E[liu] E[shi] E[hui] E[da] E[ye] E[jie] E[hun] E[SEP]
Char Embeddings
E[CLS] E[六] E[十] E[秽] E[大] E[爷] E[结] E[混] E[SEP]
Segment Embeddings
E[A] E[A] E[A] E[A] E[A] E[A] E[A] E[A] E[A]
Position Embeddings
E[0] E[1] E[2] E[3] E[4] E[5] E[6] E[7] E[8]
Input
[CLS] 六 十 秽 大 爷 结 混 [SEP]
BERT Hiddens
H[CLS] H[六] H[十] H[秽] H[大] H[爷] H[结] H[混] H[SEP]
Char Embeddings
E[CLS] E[六] E[十] E[秽] E[大] E[爷] E[结] E[混] E[SEP]
Segment Embeddings
E[A] E[A] E[A] E[A] E[A] E[A] E[A] E[A] E[A]
Position Embeddings
E[0] E[1] E[2] E[3] E[4] E[5] E[6] E[7] E[8]
Input
[CLS] 六 十 秽 大 爷 结 混 [SEP]

Figure 1. The Feature-enhanced Siamese BERT. Shape, pronunciation, token, segment, and position embeddings are summed up as input for FE-BERT in the left side. The pointer to combine BERT and FE-BERT is conditioned on CLS representation of BERT and token representations of BERT and FE-BERT.

More specifically, shape embeddings for visual features and pronunciation embeddings for phonologically features of each character are provided for the Feature-enhanced BERT as clues for selecting the best candidate. Shape embeddings of all Chinese characters are retrieved from a Chinese character glyph graph. In the glyph graph, Characters with the same components in the structure are connected and the edges are given different weights in terms of the number of character strokes and node neighbors. The pronunciation embeddings are initialized according to Chinese Pinyin, which serves as a romanization system providing phonetic-based information for Chinese characters.

In the following subsections, we describe (1) the Feature-enhanced BERT(FE-BERT) for errors similar in pronunciation or shape. (2) the combination of BERT and FE-BERT.

Algorithm 1 The Feature-enhanced Siamese BERT

```
Input: Chinese characters $X = \{x_1, x_2, \ldots, x_l\}$
Output: The corrected characters $Y = \{y_1, y_2, \ldots, y_l\}$
  Construct the glyph graph G according to structure information of Chinese characters.
  Pre-train the glyph graph G to generate the shape embedding: $E_{shape} \leftarrow Node2vec(G)$
  Randomly initialize the pronunciation embedding: $E_{pron}$
  Token embeddings $E_{token} \leftarrow Eembeds_{token}(X)$
  Segment embeddings $E_{segement} \leftarrow Eembeds_{segment}(X)$
  Position embeddings $E_{position} \leftarrow Eembeds_{position}(X)$
  $H_{fe} = \{h_{feCLS}, h_{fe1}, h_{fe2}, \ldots, h_{fel}\} \leftarrow BERT(E_{shape} + E_{token} + E_{pron} + E_{segement} + E_{position})$
  $H = \{h_{CLS}, h_1, h_2, \ldots, h_l\} \leftarrow BERT(E_{token} + E_{segement} + E_{position})$
  $P_{sem} \leftarrow Sigmoid(W_p[H; H_{fe}; H_{CLS}] + b_p)$
  $logits \leftarrow softmax(p_{sem} W_1 H + (1 - p_{sem}) W_2 H_{fe} + b_1))$
  $Y = \arg\max logits$
```

3.3. Glyph Graph

3.3.1. Shape Embeddings for Visual Features

We retrieve shape embeddings for Chinese characters from a glyph graph. As illustrated in Figure 2, Chinese characters can be repeatedly disassembled until the structure can not be subdivided. For example, character “您” can be divided into “你” and “心”, while “你” can be split into “亻” and “尔”. The same is true for character “茨”. Although the shapes of Chinese characters are different from others, they share common basic components which are also Chinese characters.

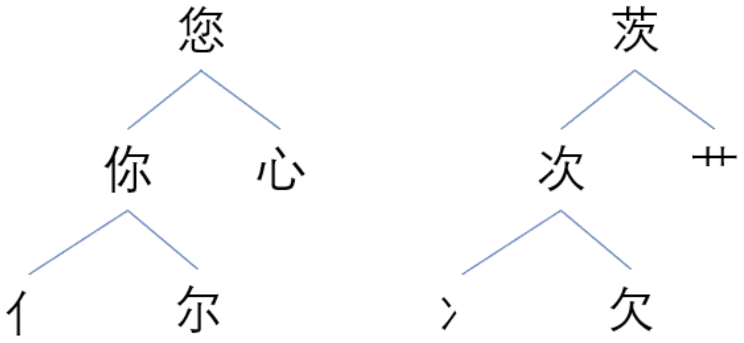


Figure 2. Chinese characters can be disassembled into smaller components, which are also Chinese characters.

The structure information of characters are labeled by the Kanji Database Project (http://kanji-database.sourceforge.net/, accessed on 12 September 2021). The shape information of Chinese characters is represented by unicode standard—Ideographic Description Sequence (IDS). For example, character “您” is represented as “U+60A8 您你心”, while character “你” is represented as “U+4F60 你亻尔”. As illustrated in Figure 3, we build the glyph graph according to IDS labels of Chinese characters by connecting characters and their sub-components. Table 2 shows some statistics of the glyph graph.

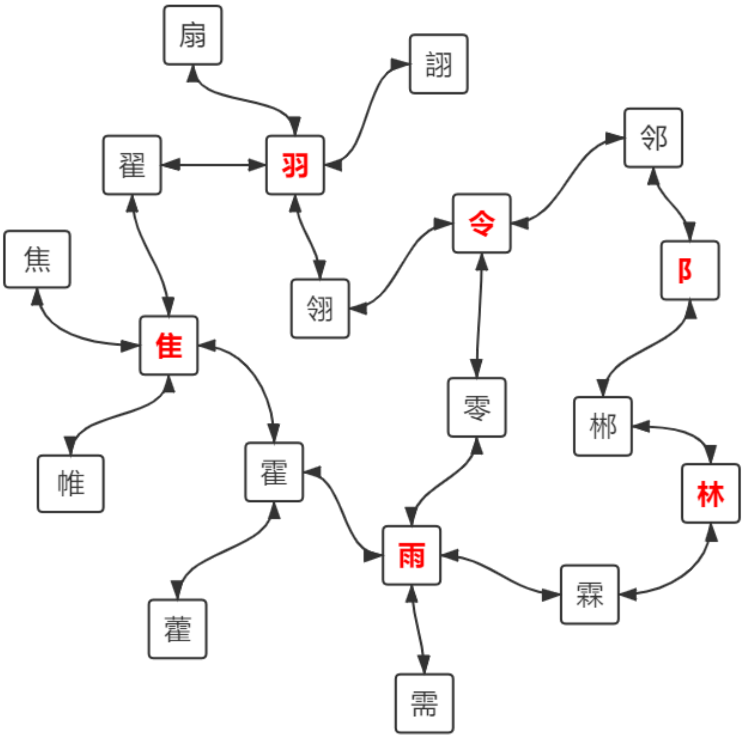


Figure 3. The glyph graph is built by connecting Chinese characters with common components.

Table 2. Statistics information of the used data resources. The number in the bracket in #Line column denotes the number of sentences with errors. For instance, 1062(526) indicates that there are a total of 1062 sentences in the test set, where contains 526 sentences with wrong words. The test set contains correct sentences to assess whether the model is correct.

Training Data	**# Line**	**Avg. Length**	**# Errors**
Auto Generated Data	269,327	42.7	378,751
SIGHAN 2013	350	49.2	350
SIGHAN 2014	6526	49.7	10,087
SIGHAN 2015	3174	30.0	4237
Total	279,377	42.7	393,425
Test Data	**# Line**	**Avg. Length**	**# Errors**
SIGHAN 2013	1000 (1000)	74.1	1227
SIGHAN 2014	1062 (526)	50.1	782
SIGHAN 2015	1100 (550)	30.5	715
Graph		**# Characters**	**# Edges**
Glyph Graph		80,627	169,745
SpellGCN: Pronunciation Graph		4753	112,687
SpellGCN: Shape Graph		4738	115,561

Considering that different character pairs have different similarity degrees and need to be distinguished, we make the glyph graph directed and weighted in terms of the number of strokes and the neighbors of graph nodes. Figure 4 shows an extreme situation that a frequently-used radical "亻" connects to 1758 characters. This indicates why the number of neighbors should be considered when setting up the edge weights. These characters sharing the same radical "亻" are similar, but their similarities should be different from each other. If every node is connected equally without weights, the characters connected by "亻" will be similar to each other with the same degree, which is not true. Besides, the number of character strokes is also important. Connected characters with closer stroke numbers may look more similar than others, such as character pair “仁” and “仨” are more similar compared to pair “亻” and “仁”.

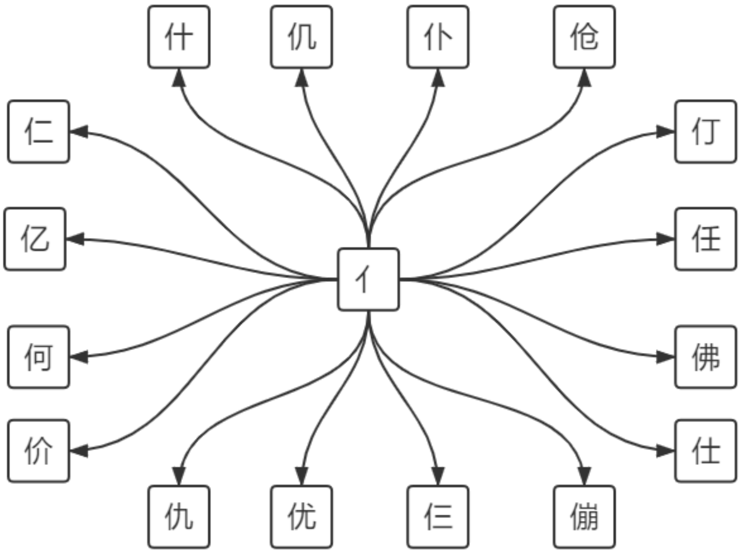


Figure 4. Some components are contained by too many other characters, such as “亻”. This figure shows a small part of 1758 characters containing “亻”.

Let $G = \{V, E, W\}$ denote the graph where $V = \{v_0, v_1, \ldots, v_N\}$ denote all Chinese characters, $w_{ij} \in W$ denotes the weight of edge $e_{ij} \in E$ pointing to character v_j from character v_i. Let s_i denote the number of strokes of character v_i, d_i denote the number of neighbors of character v_i. The edge weight w_{ij} is defined in Formula (1). The 0.5 in the denominator is to avoid dividing by zero.

$$w_{ij} = (\frac{min(s_i, s_j)}{|s_i - s_j| + 0.5} \times \frac{1}{d_j})^2 \tag{1}$$

3.3.2. Shape Embeddings

Node2vec [19] algorithm is adopted to retrieve shape representations for characters in the glyph graph. Node2vec is an extension of the Word2vec [20] algorithm, which can generate vector representations of nodes on a graph. It follows the intuition that random walks through a graph can be treated like sentences in a corpus. A random walk is treated as a sentence and each node in the walk is treated as a word. By applying Word2vec algorithms such as skip-gram or continuous bag of words model on these “sentences”, the algorithm generates shape embeddings for all nodes in the glyph graph. Visualization results of shape embeddings will be shown in Section 4.6.

It should be emphasized that the purpose of the glyph graph is not to directly learn the similarity between characters, but to learn the structure information of characters. Therefore models using these shape embeddings can learn structure relation from error pairs similar in shape and make corrections under the guidance of visual features.

3.4. Pronunciation Embeddings Phonological Features

Pinyin is an official romanization system for Standard Chinese, which provides phonetic-based information for Chinese characters. The word “Pinyin” literally means “spelled sounds”. In the Pinyin system, each character has one syllable, which consists of three components: an initial (consonant), a final (vowel), and a tone [21]. There are thousands of Chinese characters sharing hundreds (402) of Pinyin codes with different tones. Following [22], we ignore the tones here because the pronunciation of characters can already be regarded as similar if the initial (consonant) and final (vowel) of Pinyin are the same. Pinyin itself is already a mature representation of phonological features of Chinese characters. Different from shape embeddings retrieved by graph-based algorithm, the pronunciation embeddings are initialized randomly according to the Pinyin codes of characters and updated during the training phase. Case analysis results show that pronunciation embeddings can deliver phonological information and help the model to make correct predictions in phonological constraints.

3.5. Feature-Enhanced BERT

We use a Feature-enhanced BERT (FE-BERT) to deal with spelling errors with similar pronunciations or shape. Compared to original BERT, FE-BERT requires two more embeddings as input: shape embeddings for visual features and pronunciation embeddings for phonetic features. Let $E_p = \{e_{p1}, e_{p2}, \ldots, e_{pM}\}$ denote the Pinyin embeddings, $E_s = \{e_{s0}, e_{s1}, \ldots, e_{sN}\}$ denote the glyph embeddings. "M" is the number of Pinyin codes for Chinese characters' pronunciation, "N" is the number of Chinese characters.

The input sentence is converted to a sequence of embeddings that are constructed by summing up token, segment, position, shape, and pronunciation embeddings. For characters in the text sequence $X = \{x_1, x_2, \ldots, x_l\}$, BERT is used to retrieve the corresponding output representations $H_{FE} = \{h_{fe1}, h_{fe2}, \ldots, h_{fel}\}$. A visualization of this construction is shown in Figure 1.

3.6. Feature-Enhanced Siamese BERT

On one hand, the existence of external features can be the guidance to correct visually and phonologically related errors. On the other hand, the external features may be useless or even harmful when it comes to visually and phonologically irrelevant spelling errors. To reach a balance between semantic constraints and external feature constraints, and to fully leverage the reasoning capacity of the pre-trained BERT, we combine the prediction results of an original BERT and the FE-BERT through a dynamic pointer p_{sem}.

For character x_i in the input sequence, there is a purely semantic vector representation h_i from BERT and another vector representation h_{fei} mixed with external features from FE-BERT. The dynamic pointer p_{sem} to combine h_i and h_{fei} is dependent on h_i, h_{fei} and h_{cls}, where h_{cls} is the representation for token "[CLS]" from BERT. The final probability for each candidate is defined as

$$p_{sem} = Sigmoid(W_p[h_i; h_{fei}; h_{cls}] + b_p) \tag{2}$$

$$P_i = softmax(p_{sem} W_1 h_i + (1 - p_{sem}) W_2 h_{fei} + b_1)) \tag{3}$$

where W_p, W_1, W_2 and b_p, b_1 are trainable parameters. The predicted character $y_i^{'}$ at position i is the character with the max probability $y_i^{'} = \arg\max P_i$. Finally, the learning objective is to maximize the log likelihood of target characters:

$$L = \max \sum_{i=1}^{l} P(y_i|X) \tag{4}$$

4. Experiments

4.1. Datasets

SIGHAN Datasets. The training datasets are composed of three benchmark datasets [10,23,24] for CSC, which has 10 k training samples in total. The SIGHAN datasets are collected from the Chinese essay section of Test for foreigners. The corresponding test datasets from SIGHAN 2013, SIGHAN 2014, SIGHAN 2015 are used to evaluate the performance of the proposed method. The characters are converted to simplified Chinese from traditional Chinese using OpenCC (https://github.com/BYVoid/OpenCC, accessed on 11 December 2021). Following [7,8], we include additional 271K samples as the supplementary training materials, which are generated by an automatic method [7]. Statistics of the SIGHAN datasets are listed in Table 2.

Error Type Balanced Dataset. SIGHAN are real-world datasets generated by the modification of international students' compositions, they reflect the real situation in practice. However, they might not be fully capable to evaluate CSC models accurately and comprehensively because: (1) The scale of SIGHAN datasets is too small. There are only 1k sentences in the SIGHAN15 test dataset. (2) The errors in SIGHAN datasets are mainly spelling errors in similar pronunciations. Errors in similar shapes and errors which are visually and phonologically irrelevant are not included. (3) Most spelling errors in current benchmark datasets are leaked to CSC models through widely-used auto-generated datasets in training process. The ability to handle unseen new errors of CSC models is never evaluated. To handle these issues, we construct a relatively large-scale Error Type Balanced Dataset.

We repartition the auto-generated 271K samples dataset and modify the spelling errors to build an error-type balanced dataset. There are four kinds of errors defined by character's similarity: similar only in shape, similar only in pronunciation, similar in both shape and pronunciation, and not similar in shape nor pronunciation. Besides, it is also necessary to evaluate errors that are not included in the confusion set. If character x_i is misspelled as x_j in training materials, the error pair $< x_i \rightarrow x_j >$ is defined as "seen" by CSC models in training phase. In the test dataset, there are spelling errors not seen by CSC models in the training materials. This is to evaluate the performance on new errors. Statistics of the Error Type Balanced Dataset are listed in Table 3.

Table 3. Statistics information of the Error Type Balanced Dataset. "Seen Errors" means the errors concluded in the training dataset. "New Errors" means the spelling errors in sentences that are not in the training dataset, which are new to CSC models. Besides, each type of error is divided into two situations according to whether the error pair appears in the confusion set. "Confusion Set Coverage" means the portion of error pairs covered by the confusion set.

Test Data				
Seen Errors				
Error Type	#phonetic	#shape	#phonetic and shape	#other
Num	2k	2k	2k	2k
Confusion Set Coverage	50%	50%	50%	50%
New Errors				
Error Type	#phonetic	#shape	#phonetic and shape	#other
Num	2k	2k	2k	2k
Confusion Set Coverage	50%	50%	50%	50%
Training Data				
Error Type	#phonetic	#shape	#phonetic and shape	#other
Num	32k	32k	32k	32k
Confusion Set Coverage	50%	50%	50%	50%

4.2. Baselines

We compare our method with several strong baselines.

- PN [7]: This method copies candidate characters from a confusion set by Pointer Network [25].
- FASpell [11]: FASPell is a new paradigm for CSC which consists of a denoising autoencoder (DAE) and a decoder. Candidate characters are retrieved by a pre-trained

masked language model and a specialized decoder utilizing the salient feature of Chinese character similarity is used to select the best candidate.

- BERT-Embed [9]: The word embeddings are used as the softmax layer on the top of BERT for the CSC task. This method served as a baseline in [9].
- BERT-Linear [9]: The original BERT without shape or phonological features. A pre-trained linear layer is used to make predictions over hidden states of the last layer of BERT.
- Spell-GCN [8]: Phonological and visual similarity knowledge is integrated into language models for CSC via a specialized graph convolutional network (SpellGCN).
- Soft-Masked BERT [12]: This model consists of a network for error detection and a network for error correction based on BERT, with the former being connected to the latter with a "soft-masking" technique.

4.3. Hyper-Parameters

Our implementation is based on the repository of pytorch-transformers (https://github.com/huggingface/pytorch-transformers, accessed on 12 November 2021). The BERT model pre-trained by huggingface (https://huggingface.co/, accessed on 12 November 2021) is used in our experiments. We fine-tune the models using AdamW [26] optimizer with Stochastic Weight Averaging (SWA) [27] for 3 epochs with a batch size of 32 and a learning rate of 5×10^{-5}.

When training shape embeddings using Node2vec algorithm on the glyph graph, the return parameter p and the in-out parameter q are 1 and 8, respectively; the length and number of walk per source are 3 and 300; the context size for optimization is 2. We force the Node2vec algorithm to concentrate on neighbors which are characters with the same components by setting the context size and walk length small and the in-out parameter q large.

4.4. Main Results

Table 4 shows experimental results of the above methods on three SIGHAN datasets. The FES-BERT equipped with shape and phonic features achieves better performance against vanilla BERT and SpellGCN on most metrics of SIGHAN datasets. In terms of sentence-level F1 score metric in the correction subtask, i.e., C-F score in the last column, the improvements against previous best results (Spell-GCN) are 3.3%, 1.5%, and 0.2% points respectively. In terms of character-level F1 score metric in the correction subtask, the improvements against SpellGCN are 2.0%, 0.1%, and −0.1%, respectively. This demonstrates the effectiveness of our proposed method on benchmark datasets.

Table 5 exhibits model performance on different kinds of spelling errors, which are respectively about pronunciation, shape, and others. There are four situations for a CSC model to correct a spelling error according to whether the error pairs are in the confusion set and whether the error pairs have been seen in training materials. For errors in both train and test datasets, SpellGCN and FES-BERT reach close scores on the sentence level F1 metrics while the latter has a small advantage. For new errors that only exist in the test dataset, the FES-BERT achieves a large advantage on the sentence level F1 metrics. This demonstrates that FES-BERT has a better generalization performance than SpellGCN and BERT. For errors that are new but included in the confusion set, SpellGCN achieves an obvious advantage over vanilla BERT. This indicates that it is helpful to collect error characters into a confusion set.

Table 4. The performance of our method and baseline models (%). D, C denote the detection, correction, respectively. P, R, F denote the precision, recall and F1 score, respectively. Best results are in **bold**. Following experiment setting of [8], We performed additional fine-tuning on SIGHAN13 for 3 epochs as the data distribution in SIGHAN13 differs from other datasets, e.g., “的得地” are rarely distinguished. Dataset. The results with ‡ are reproduced by our own implementation.

	Character-Level						**Sentence-Level**					
	Detection-Level			**Correction-Level**			**Detection-Level**			**Correction-Level**		
SIGHAN 2013	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	C-P	C-R	C-F
LMC [28]	79.8	50.0	61.5	77.6	22.7	35.1	(-)	(-)	(-)	(-)	(-)	(-)
SL [3]	54.0	69.3	60.7	(-)	(-)	52.1	(-)	(-)	(-)	(-)	(-)	(-)
PN [7]	56.8	91.4	70.1	79.7	59.4	68.1	(-)	(-)	(-)	(-)	(-)	(-)
FASpell [11]	(-)	(-)	(-)	(-)	(-)	(-)	76.2	63.2	69.1	73.1	60.5	66.2
BERT-Embed [8]	80.6	88.4	84.3	98.1	87.2	92.3	79.0	72.8	75.8	77.7	71.6	74.6
BERT-Linear	83.7	87.9	85.7	**98.6**	86.7	92.3	83.4	76.5	79.8	**82.0**	75.2	78.4
Soft-Masked BERT [12] ‡	**84.9**	87.6	86.2	97.5	85.4	91.1	**84.0**	77.2	**80.4**	81.5	74.8	78.0
SpellGCN [8]	82.6	88.9	85.7	98.4	**88.4**	**93.1**	80.1	74.4	77.2	78.3	72.7	75.4
FES-BERT	84.4	**89.6**	**86.9**	98.4	88.2	93.0	82.9	**77.7**	80.2	81.3	**76.2**	**78.6**
SIGHAN 2014	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	C-P	C-R	C-F
LMC [28]	56.4	34.8	43.0	71.1	50.2	58.8	(-)	(-)	(-)	(-)	(-)	(-)
SL [3]	51.9	66.2	58.2	(-)	(-)	56.1	(-)	(-)	(-)	(-)	(-)	(-)
PN [7]	63.2	82.5	71.6	79.3	68.9	73.7	(-)	(-)	(-)	(-)	(-)	(-)
FASpell [11]	(-)	(-)	(-)	(-)	(-)	(-)	61.0	53.5	57.0	59.4	52.0	55.4
BERT-Embed [8]	82.9	77.6	80.2	96.8	75.2	84.6	65.6	68.1	66.8	63.1	65.5	64.3
BERT-Linear	82.3	78.1	80.1	96.6	75.4	84.7	66.4	67.8	67.1	64.0	65.3	64.7
Soft-Masked BERT [12] ‡	**84.1**	77.5	80.6	95.4	73.9	83.3	66.9	69.7	68.3	63.8	66.5	65.1
SpellGCN [8]	83.6	78.6	**81.0**	**97.2**	76.4	85.5	65.1	69.5	67.2	63.1	67.2	65.3
FES-BERT	83.9	**78.7**	**81.3**	97.1	**76.5**	**85.6**	**67.6**	**70.1**	**68.8**	**65.6**	**68.0**	**66.8**
SIGHAN 2015	D-P	D-R	D-F	C-P	C-R	C-F	D-P	D-R	D-F	C-P	C-R	C-F
LMC [28]	83.8	26.2	40.0	71.1	50.2	58.8	(-)	(-)	(-)	(-)	(-)	(-)
SL [3]	56.6	69.4	62.3	(-)	(-)	57.1	(-)	(-)	(-)	(-)	(-)	(-)
PN [7]	66.8	73.1	69.8	71.5	59.5	69.9	(-)	(-)	(-)	(-)	(-)	(-)
FASpell [11]	(-)	(-)	(-)	(-)	(-)	(-)	67.6	60.0	63.5	66.6	59.1	62.6
BERT-Embed [8]	87.5	85.7	86.6	95.2	81.5	87.8	73.7	78.2	75.9	70.9	75.2	73.0
BERT-Linear	86.7	83.6	85.1	94.3	78.9	85.9	76.6	78.0	77.3	73.6	74.9	74.2
Soft-Masked BERT [12] ‡	88.0	84.5	86.2	94.9	80.1	86.9	77.5	79.1	78.3	74.5	76.0	75.2
SpellGCN [8]	88.9	**87.7**	**88.3**	95.7	83.9	89.4	74.8	80.7	77.7	72.1	77.7	75.9
FES-BERT	**89.4**	87.3	**88.3**	**97.1**	**84.8**	**90.5**	**79.6**	**82.2**	**80.9**	**78.0**	**80.5**	**79.2**

Table 5. Sentence level F1 scores on Error Type Balanced Dataset. “In” means in the confusion set, “out” means out of the confusion set, “p” means phonic, “s” means shape. “New errors” are error pairs never seen by CSC models in the training phase while “seen errors” are error pairs appeared in training materials; “all” in last column means sentence level F1 scores on all error types together.

	New Errors								
	in_p	in_s	in_sp	in_other	out_p	out_s	out_sp	out_other	all
FES-BERT	**31.45**	**47.35**	**38.46**	**45.11**	**64.67**	**54.53**	**61.55**	**53.52**	**49.96**
Spell-GCN	24.61	45.11	31.18	40.16	58.43	49.11	49.68	49.92	43.96
BERT-Linear	21.05	45.69	27.89	39.54	55.04	49.36	49.03	53.2	43.02
Soft-Masked BERT	21.03	45.25	28.65	37.85	53.47	49.84	48.61	51.52	42.51

Table 5. *Cont.*

Seen Errors									
	in_p	in_s	in_sp	in_other	out_p	out_s	out_sp	out_other	all
FES-BERT	**73.86**	**91.85**	**77.33**	93.29	**91.48**	**90.4**	**96.94**	**92.78**	**88.69**
Spell-GCN	71.54	88.21	75.66	**93.95**	88.28	88.25	96	91.17	86.76
BERT	69.76	89.14	71.86	93.85	86.57	87.9	94.83	90.59	85.8
Soft-Masked BERT	69.29	88.37	70.08	92.98	81.95	86.18	91.22	89.13	83.87

4.5. Ablation Studies

In this section, we analyze the effect of shape embeddings and pronunciation embeddings on SIGHAN15. We trained FES-BERT with only pronunciation embeddings or shape embeddings as an external feature. The experimental results on SIGHAN15 can be seen in Table 6. Figure 5 shows the test curves of FES-BERT-S, FES-BERT-P, FES-BERT, BERT-Linear, FE-BERT, and Soft-Masked BERT.

Compared to vanilla BERT, the FES-BERT with additional features converges rapidly in less than four epochs. In terms of sentence-level F1 score metric in the correction subtask at epoch 3, the improvements of FES-BERT-S, FES-BERT-P, and FE-BERT against the original BERT (BERT-Linear) are 2.9%, 3.3%, and 1.8%, respectively. In terms of character-level F1 score metric in the detection subtask, the improvements of FES-BERT-S, FES-BERT-P, and FE-BERT against the original BERT are 3.9%, 4.7%, and 1.8%, respectively. All three models achieve better performance on all metrics than to the original BERT. This indicates that both pronunciation and shape features are necessary to correct Chinese spelling errors. Models with the siamese structure achieve higher scores than FE-BERT, which means the siamese structure is also important for correcting spelling errors.

Table 6. Experimental results of ablation study. FES-BERT-P means FES-BERT with only pronunciation features, FES-BERT-S means FES-BERT with only shape features. FE-BERT is a Feature-enhanced BERT with shape and pronunciation features but without the siamese original BERT.

	Character-Level						Sentence-Level					
	Detection-Level			Correction-Level			Detection-Level			Correction-Level		
SIGHAN 2015	**D-P**	**D-R**	**D-F**	**C-P**	**C-R**	**C-F**	**D-P**	**D-R**	**D-F**	**C-P**	**C-R**	**C-F**
FES-BERT-P	88.7	87.4	88.0	97.1	84.9	90.6	77.7	80.9	79.3	75.9	79.1	77.5
FES-BERT-S	86.6	85.3	86.0	97.5	83.2	89.8	78.9	78.0	78.4	77.6	76.7	77.1
FE-BERT	86.4	83.4	84.8	96.5	80.4	87.7	78.3	77.6	78.0	76.3	75.6	76.0

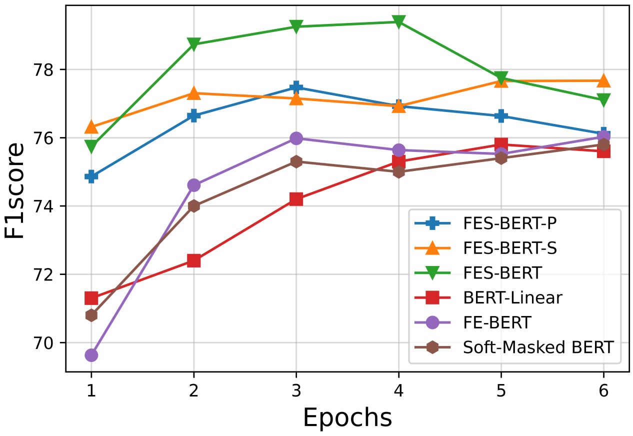


Figure 5. The test curves for sentence-level correction metrics of FES-BERT, FES-BERT-S, FES-BERT-P, FE-BERT, Soft-Masked BERT, and BERT.

4.6. Case Study

We show several spelling error cases either similar in shape or similar in pronunciation in Table 7. There are more than one semantically appropriate candidate characters for these cases, which means information of pronunciation and shape is necessary to make further judgments.

For sentences with error characters in similar shapes, such as “他们计划舂(chong)天去爬山。”, the corresponding correct sentence is “他们计划春(chun)天去爬山。(They plan to climb mountains in spring.)”. The Chinese character “ 春(chun)” is misspelled as “ 舂(chong)” in a similar shapes. There are several semantically appropriate candidate characters such as “秋(qiu)”, “春(chun)” and “明(ming)”. It is reasonable in semantics for BERT to correct the sentence to “他们计划明(ming)天去爬山。(They plan to climb mountains tomorrow.)” or “他们计划秋(qiu)天去爬山。(They plan to climb and mountains in autumn.)”. However, both 明(ming) and 秋(qiu) are not consistent with the shape constraints of 舂(chong). The errors are easy to detect but difficult to correct for vanilla BERT with only semantic information. However, the FES-BERT can find the best candidates “春(chun)” according to the shape constraints from shape embeddings of character “舂(chong)”. The same is true for character pair “愁(chou)” and “秋(qiu)”. SpellGCN also makes correct predictions in the case of “舂” and “春“ which are in the character pairs of the confusion set. However, SpellGCN fails to correct “轻哽(geng)卡车” to “轻便(bian)卡车” because character pair “哽(geng)” and “便(bian)” are not in the confusion set, which means “哽(geng)” and “便(bian)” are considered similar by Spell-GCN.

For sentences with error characters in similar pronunciations, FES-BERT can also make correct predictions according to phonic features. For example, in sentence “围这(zhe)他的摄影师将近二百三十人。”, “这(zhe)” is an error character with similar pronunciations but different shapes compared to “着(zhe)”. FES-BERT corrects “围这(zhe)” to “围着(zhe)”, while BERT and SpellGCN corrects “围这(zhe)” to “围攻(gong)” and “围堵(du)” according to semantic constraints. For spelling errors “埔(bu)办” and “琥(hu)区”, both FES-BERT and SpellGCN predict correct answers “补(bu)办” and “虎(hu)区” according to character pronunciation while the original BERT makes error predictions “举(ju)办” and “误(wu)区” , which are misled by the semantic context.

The Soft-Masked BERT corrects both “舂(chong)天” and “愁(chou)天” into “秋(qiu)天”. The reason may be that the soft masks avoids the influence of original error characters. Besides, it is worth mentioning that most spelling errors can be corrected by BERT according to semantic information without other features. If the error pair in input sentence has appeared in the training materials, BERT can make correct predictions easily.

Table 7. Sentences with spelling errors similar in shape or pronunciation. The code in brackets is the Pinyin representation for the corresponding character. The incorrect/correct characters are in orange/blue.

Error Sentence	BERT	FES-BERT	SpellGCN	Soft-Masked BERT
Characters With Similar Shapes				
他们计划舂(chong)天去爬山。	明(ming)	春(chun)	春(chun)	秋(qiu)
所谓轻哽(geng)卡车包括小卡车、小货车以及休旅车。	型(xing)	便(bian)	型(xing)	型(xing)
经过膦(lin)选，他成为了公务员。	民(min)	遴(lin)	遴(lin)	遴(lin)
马来西亚表示已斩(zan)定明年一月中旬的某日遣返米苏阿里。	预(yu)	暂(zan)	预(yu)	预(yu)
遭擂(lei)击巴士的驾驶罹难，不过卡车司机幸存。	攻(gong)	撞(zhuang)	攻(gong)	攻(gong)
他们计划愁(chou)天去爬山。	明(ming)	秋(qiu)	明(ming)	秋(qiu)

Table 7. *Cont.*

Error Sentence	BERT	FES-BERT	SpellGCN	Soft-Masked BERT
Characters with Similar Pronunciations				
围这(zhe)他的摄影师将近二百三十人。	攻(gong)	着(zhe)	堵(du)	攻(gong)
我们对演尝(chang)会的事感到很高兴	奏(zou)	唱(chang)	讲(jiang)	唱(chang)
原则同意为项目埔(bu)办手续。	举(ju)	补(bu)	补(bu)	申(shen)
所以间(jian)管的时间将不再仅仅只局限于新股首日表现。	尽(jin)	监(jian)	尽(jin)	间(jian)
一只误入琥(hu)区的小黑熊被东北虎围攻咬死。	误(wu)	虎(hu)	虎(hu)	虎(hu)
为冬季残奥会提供残嫉(ji)人设施。	障(zhang)	疾(ji)	障(zhang)	障(zhang)

4.7. Visualization of Embeddings

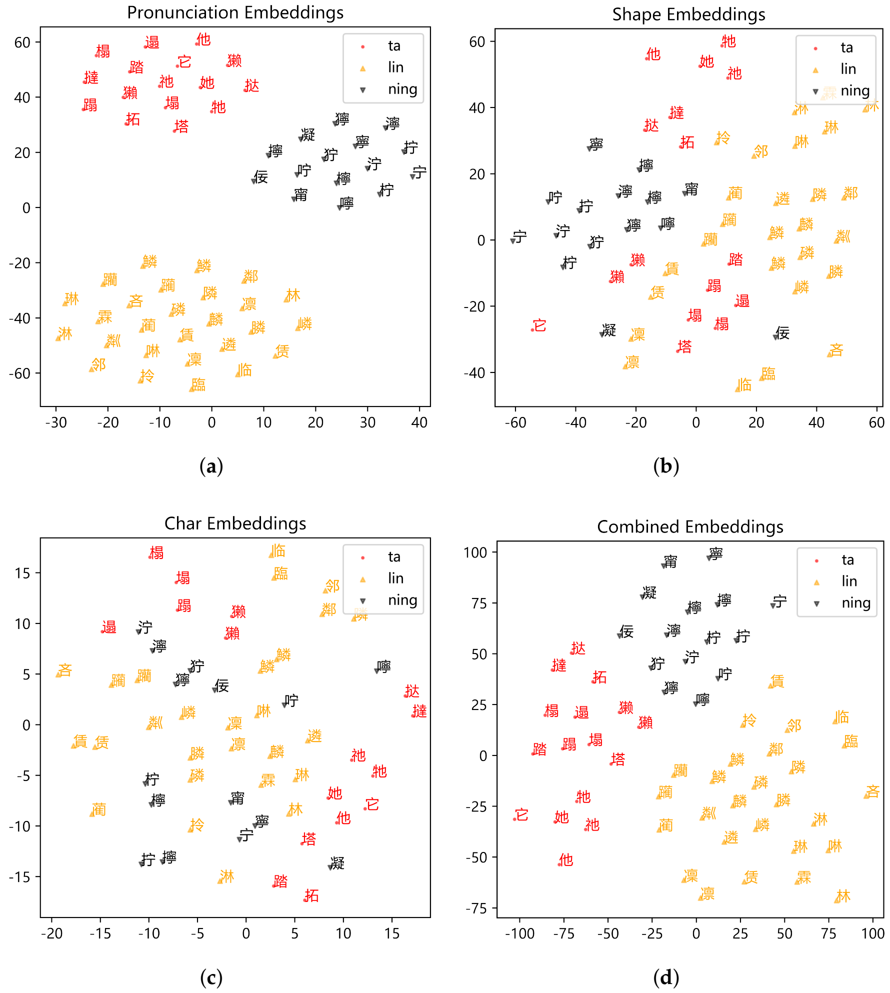
Figure 6 shows visualization results of embedddings by t-SNE [29]. Characters with Pinyin "lin", "ning", or "ta" are presented in the figure.

Figure 6a shows visualization results of pronunciation embeddings. Characters with different pronunciations are separated easily. The reason is that characters with similar pronunciations in the graph share the same Pinyin codes (without tone), so their phonological features are represented by the same pronunciation embedding.

Figure 6b shows visualization results of shape embeddings, in which the similar characters in terms of shape are placed together. For example, characters "宁柠拧泞咛" with common components "宁" are located closer to each other than other characters. This indicates the shape embeddings retrieved from the glyph graph do contain structure information of Chinese characters and can be used to measure the similarity of characters in shape.

Figure 6c shows visualization results of BERT token embeddings. Characters in Figure 6c do not gather together either by shape or sound. Embeddings of original BERT failed to capture similarity information in terms of shape or pronunciation.

Figure 6d shows visualization results of embeddings which are the sum of shape, pronunciation, and BERT char embeddings. In Figure 6d, characters with similar shapes and similar pronunciations exhibit cluster patterns in an obvious trend. The characters with similar shapes such as "遴磷麟鳞膦" or "他她地" are placed closely from each other. Characters with similar pronunciations exhibit the same phenomenon, are also closely. Characters with both similar shapes and similar pronunciations are placed closer than those with only one similar feature. For example, "遢榻拓" are characters with the same pronunciation, so these three characters are closer than other words. Meanwhile, "遢榻" looks more similar than "塌拓", so "遢榻" are closer than "塌拓" in Figure 6d. Due to this property, the feature-enhanced model may tend to recognize the similarity between characters and is able to search for answers with shape or pronunciation constraints.



(**a**)

(**b**)

(**c**)

(**d**)

Figure 6. Scatters of embeddings reduced by t-SNE. The colors in the picture indicate the pronunciation of Chinese characters, e.g., red is for Pinyin code "ta", orange in for Pinyin code "lin", black is for Pinyin code "ning". (**a**) Scatter of characters in terms of pronunciation embeddings. (**b**) Scatter of characters in terms of shape embeddings. (**c**) Scatter of characters in terms of char embeddings of BERT. (**d**) Scatter of characters in terms of the sum of pronunciation, shape, char embeddings.

5. Conclusions

We incorporate shape and pronunciation features of Chinese characters into BERT language models, and a good balance between semantic constraints and Chinese characters' external feature constraints is reached via a siamese model structure. A novel and efficient graph-based method is introduced for retrieving shape features of Chinese characters. The proposed method achieves better experimental results compared to previous SOTA models SpellGCN and Soft-Masked BERT. Case studies compared to the original BERT, SpellGCN, and Soft-Masked BERT show that our model is able to search candidate characters more accurately with the constraints of shape and pronunciation. For more effective development and evaluation of CSC methods, we built an error type balanced dataset by repartitioning the auto-generated 271K CSC dataset and modifying the types of spelling errors.

As for future work, we plan to develop an end-to-end CSC system based on the glyph graph and explore models with more powerful reasoning capabilities through external features of characters, instead of simply "remembering" spelling errors that appeared in the training materials.

Author Contributions: Funding acquisition, T.W.; Investigation, H.G.; Methodology, Y.L.; Software, S.W.; Validation, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 22008036.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yu, J.; Li, Z. Chinese Spelling Error Detection and Correction Based on Language Model, Pronunciation, and Shape. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014; Sun, L., Zong, C., Zhang, M., Levow, G., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 220–223. [CrossRef]
2. Zhang, S.; Xiong, J.; Hou, J.; Zhang, Q.; Cheng, X. HANSpeller++: A Unified Framework for Chinese Spelling Correction. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, 30–31 July 2015; Yu, L., Sui, Z., Zhang, Y., Ng, V., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp. 38–45. [CrossRef]
3. Wang, D.; Song, Y.; Li, J.; Han, J.; Zhang, H. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2517–2527. [CrossRef]
4. Xu, J.; Tang, B.; He, H.; Man, H. Semisupervised Feature Selection Based on Relevance and Redundancy Criteria. *IEEE Trans. Neural Networks Learn. Syst.* **2017**, *28*, 1974–1984. [CrossRef] [PubMed]
5. Yang, X.; Liao, L.; Yang, Q.; Sun, B.; Xi, J. Limited-energy output formation for multiagent systems with intermittent interactions. *J. Frankl. Inst.* **2021**, *358*, 6462–6489. [CrossRef]
6. Liao, D.; Xu, J.; Li, G.; Huang, W.; Liu, W.; Li, J. Popularity Prediction on Online Articles with Deep Fusion of Temporal Process and Content Features. In Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, HI, USA, 27 January–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; pp. 200–207. [CrossRef]
7. Wang, D.; Tay, Y.; Zhong, L. Confusionset-guided Pointer Networks for Chinese Spelling Check. In Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, 28 July–2 August 2019, Volume 1: Long Papers; Korhonen, A., Traum, D.R., Màrquez, L., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 5780–5785. [CrossRef]
8. Cheng, X.; Xu, W.; Chen, K.; Jiang, S.; Wang, F.; Wang, T.; Chu, W.; Qi, Y. SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, online, 6 August 2021; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 871–881. [CrossRef]
9. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; Volume 1 (Long and Short Papers); Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 4171–4186. [CrossRef]
10. Yu, L.; Lee, L.; Tseng, Y.; Chen, H. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014; pp. 126–132. [CrossRef]
11. Hong, Y.; Yu, X.; He, N.; Liu, N.; Liu, J. FASPell: A Fast, Adaptable, Simple, Powerful Chinese Spell Checker Based On DAE-Decoder Paradigm. In Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019), Hong Kong, China, 4 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 160–169. [CrossRef]
12. Zhang, S.; Huang, H.; Liu, J.; Li, H. Spelling Error Correction with Soft-Masked BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 882–890. [CrossRef]
13. Liu, X.; Cheng, K.; Luo, Y.; Duh, K.; Matsumoto, Y. A Hybrid Chinese Spelling Correction Using Language Model and Statistical Machine Translation with Reranking. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, 14–18 October 2013; Yu, L., Tseng, Y., Zhu, J., Ren, F., Eds.; Asian Federation of Natural Language Processing: Nagoya, Japan, 2013; pp. 54–58.
14. Zheng, B.; Che, W.; Guo, J.; Liu, T. Chinese Grammatical Error Diagnosis with Long Short-Term Memory Networks. In Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications, NLP-TEA@COLING 2016, Osaka, Japan, 12 December 2016; Chen, H., Tseng, Y., Ng, V., Lu, X., Eds.; The COLING 2016 Organizing Committee: Osaka, Japan 2016; pp. 49–56.

15. Yang, Y.; Xie, P.; Tao, J.; Xu, G.; Li, L.; Luo, S. Alibaba at IJCNLP-2017 Task 1: Embedding Grammatical Features into LSTMs for Chinese Grammatical Error Diagnosis Task. In Proceedings of the IJCNLP 2017, Shared Tasks, Taipei, Taiwan, 27 November–1 December 2017; Liu, C., Nakov, P., Xue, N., Eds.; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 41–46.
16. Wu, S.; Wang, J.; Chen, L.; Yang, P. CYUT-III Team Chinese Grammatical Error Diagnosis System Report in NLPTEA-2018 CGED Shared Task. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications, NLP-TEA@ACL 2018, Melbourne, Australia, 19 July 2018; Tseng, Y., Chen, H., Ng, V., Komachi, M., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 199–202. [CrossRef]
17. Bao, Z.; Li, C.; Wang, R. Chunk-based Chinese Spelling Check with Global Optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 2031–2040. [CrossRef]
18. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. In Proceedings of the NIPS 2014 Workshop on Deep Learning, Montreal, QC, Canada, 13 December 2014.
19. Grover, A.; Leskovec, J. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 855–864. [CrossRef]
20. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, AZ, USA, 2–4 May 2013.
21. Jiang, Z.; Gao, Z.; He, G.; Kang, Y.; Sun, C.; Zhang, Q.; Si, L.; Liu, X. Detect Camouflaged Spam Content via StoneSkipping: Graph and Text Joint Embedding for Chinese Character Variation Representation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 6187–6196. [CrossRef]
22. Zhang, R.; Pang, C.; Zhang, C.; Wang, S.; He, Z.; Sun, Y.; Wu, H.; Wang, H. Correcting Chinese Spelling Errors with Phonetic Pre-training. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, 1–6 August 2021*; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Stroudsburg, PA, USA, 2021; pp. 2250–2261. Volume ACL/IJCNLP 2021. [CrossRef]
23. Wu, S.H.; Liu, C.L.; Lee, L.H. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, 14–18 October 2013; Asian Federation of Natural Language Processing: Nagoya, Japan, 2013; pp. 35–42.
24. Tseng, Y.H.; Lee, L.H.; Chang, L.P.; Chen, H.H. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, 30–31 July 2015; Association for Computational Linguistics: Beijing, China, 2015; pp. 32–37. [CrossRef]
25. Gulcehre, C.; Ahn, S.; Nallapati, R.; Zhou, B.; Bengio, Y. Pointing the Unknown Words. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 140–149. [CrossRef]
26. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net, 2019.
27. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A. Averaging weights leads to wider optima and better generalization. In Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, CA, USA, 6–10 August 2018; Silva, R., Globerson, A., Globerson, A., Eds.; Association For Uncertainty in Artificial Intelligence (AUAI): Monterey, CA, USA, 2018; pp. 876–885.
28. Xie, W.; Huang, P.; Zhang, X.; Hong, K.; Huang, Q.; Chen, B.; Huang, L. Chinese Spelling Check System Based on N-gram Model. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, 30–31 July 2015; pp. 128–136. [CrossRef]
29. Rauber, P.E.; Falcão, A.X.; Telea, A.C. Visualizing Time-Dependent Data Using Dynamic t-SNE. In Proceedings of the 18th Eurographics Conference on Visualization, EuroVis 2016—Short Papers, Groningen, The Netherlands, 6–10 June 2016; Bertini, E., Elmqvist, N., Wischgoll, T., Eds.; Eurographics Association: Goslar, Germany, 2016; pp. 73–77. [CrossRef]