Dynamic Connected Networks for Chinese Spelling Check

Baoxin Wang[1,2], **Wanxiang Che**[1], **Dayong Wu**[2], **Shijin Wang**[2,3], **Guoping Hu**[2], **Ting Liu**[1]
[1]Research Center for SCIR, Harbin Institute of Technology, Harbin, China
[2]State Key Laboratory of Cognitive Intelligence, iFLYTEK Research, China
[3]iFLYTEK AI Research (Hebei), Langfang, China
{bxwang2,dywu2,sjwang3,gphu}@iflytek.com
{car,tliu}@ir.hit.edu.cn

Abstract

Chinese spelling check (CSC) is a task to detect and correct spelling errors in Chinese text. Most state-of-the-art works on the CSC task adopt a BERT-based non-autoregressive language model, which relies on the output independence assumption. The inappropriate independence assumption prevents BERT-based models from learning the dependencies among target tokens, resulting in an incoherent problem. To address the above issue, we propose a novel architecture named Dynamic Connected Networks (DCN), which generates the candidate Chinese characters via a Pinyin Enhanced Candidate Generator and then utilizes an attention-based network to model the dependencies between two adjacent Chinese characters. The experimental results show that our proposed method achieves a new state-of-the-art performance on three human-annotated datasets.

1 Introduction

Chinese spelling check (CSC) is an important task which can be utilized in many natural language applications such as optical character recognition (OCR) (Wang et al., 2018; Hong et al., 2019) and essay scoring. Meanwhile, CSC is a challenging task which requires human-level natural language understanding ability (Liu et al., 2010, 2013; Xin et al., 2014). Recently, BERT-based non-autoregressive language models have achieved state-of-the-art performance in the CSC task (Hong et al., 2019; Zhang et al., 2020; Cheng et al., 2020).

These works fine-tune BERT-based models using CSC training data. During the training phase, all the target Chinese characters will be involved as labels. In the inference stage, the models predict the most likely Chinese character from a candidate set at each position. When the most likely character is different from the input character, the input character will be considered as a spelling error and corrected to the most likely character. Based on the powerful generalization ability of BERT (Devlin et al., 2019), these works have achieved better performance than other models.

Wrong: 我忘记告诉你了，我真户秃。
Correct: 我忘记告诉你了，我真糊涂。 Translation: I forgot to tell you. I'm so confused.

Table 1: An example of Chinese spelling errors. Here, "户秃" should be corrected to "糊涂" (confused).

However, these works on the CSC task rely on the incorrect independence assumption, which may lead to an incoherent problem. Concretely, they assume that the predicted tokens are independent of each other, which generally does not hold in natural language (Yang et al., 2019; Gu and Kong, 2020). For the CSC task, one spelling error may have multiple corrections. Ignoring the corrected context may result in a correction conflict. As shown in Table 1, "户秃" may be corrected as "糊涂" (*confused*) or "尴尬" (*embarrassed*). Because of the independence of each token, the non-autoregressive language model may correct it as "尴涂" (*embafused*). This incoherent problem is also called a multi-modality problem in non-autoregressive machine translation (Gu et al., 2018).

To address the above problem, we propose a novel Dynamic Connected Networks (DCN) which can model the dependencies between two adjacent candidate Chinese characters. Specifically, we use the RoBERTa model (Liu et al., 2019; Cui et al., 2019) as our base model, which can also be replaced by other models. Firstly, we utilize RoBERTa with a Pinyin Enhanced Candidate Generator to incorporate phonological information and generate k candidate characters at each position. For each two adjacent candidates, DCN learns a

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2437–2446

variable connection score to determine the strength of the dependency between them via a Dynamic Connected Scorer (DCScorer). The DCScorer calculates the connection scores by feeding the contextual representation and the candidate character embeddings of the current and the next position into an attention layer simultaneously. Eventually, the model generates k^n candidate paths, and we utilize the Viterbi algorithm (Rabiner, 1989) to quickly find the one with the highest score as our final correction result.

Conditional random fields (CRF) (Lafferty et al., 2001) can also model the dependencies of output labels, however it is not suitable for language modeling or the CSC task. The dependencies between Chinese characters are more related to the context and far more complicated than the label relations of other tasks such as NER. Thus, the capacity of a fixed transition matrix in CRF is inadequate. Moreover, the number of Chinese characters is usually more than 5K, making the transition matrix too large to learn. In contrast, output candidates (labels) and connection scores of DCN are dynamic and change according to the context. That empowers our model with a strong ability to learn the dependencies.

We conduct experiments on SIGHAN 2013, SIGHAN 2014, and SIGHAN 2015 benchmarks. Experimental results on the three human-annotated datasets demonstrate that the performance of our proposed method is significantly better than the state of the art models.

To summarize, our contributions are as follows:

- We propose a novel end-to-end dynamic connected networks (DCN) which can alleviate the incoherent problem of non-autoregressive language models in the CSC task.
- We propose a simple and effective Pinyin Enhanced Candidate Generator to incorporate phonological information and generate better candidate characters.
- Experimental results show that our proposed method achieves state-of-the-art performance on three human-annotated datasets.

For reproducibility, our code for this paper is available at https://github.com/destwang/DCN.

2 Related Work

Chinese spelling check (CSC) is a challenging task that requires human-level language understanding ability. With the development of deep learning techniques, the CSC task has recently made more progress. CSC is similar to the grammatical error correction (GEC) task (Dahlmeier and Ng, 2012). The difference between them is that CSC only focuses on Chinese spelling errors, while GEC also includes errors that need insertion and deletion.

Most models in the GEC task use an autoregressive Seq2Seq model to correct a sentence. Similarly, Seq2Seq models can also be used in the CSC task. Wang et al. (2019) propose an autoregressive pointer network which generates a Chinese character from the confusion set rather than the entire vocabulary. Although the autoregressive Seq2Seq model has the ability to correct the spelling errors, it is usually slow. The input and output are so similar that it would be "wasteful" to completely regenerate a sequence (Malmi et al., 2019).

Since the input and output have the same number of Chinese characters, and the correct and incorrect Chinese characters correspond to each other, it is more intuitive to use non-autoregressive language models such as BERT to directly correct the Chinese spelling errors. Hong et al. (2019) propose the FASPell model to predict candidate characters based on the BERT model and exploit the phonological and visual similarity information to select candidate characters. Zhang et al. (2020) propose a model named Soft-Masked BERT, which consists of a detection network and a correction network based on BERT. Cheng et al. (2020) propose to incorporate phonological and visual similarity knowledge into BERT via a specialized graph convolutional network. Bao et al. (2020) design a chunk-based framework and extend the traditional confusion sets with semantical candidates to cover different types of errors.

Although these non-autoregressive methods mentioned above have achieved state of the art in the CSC task so far, these methods still suffer from the incoherent problems that exist in non-autoregressive models (Gu et al., 2018; Gu and Kong, 2020). In this paper, we propose a novel model DCN which learns the dependencies between the adjacent Chinese characters and alleviates the incoherent problem.

3 Our approach

3.1 Problem

Given an input text sequence $X = \{x_1, x_2, ..., x_N\}$, the goal of the CSC task is

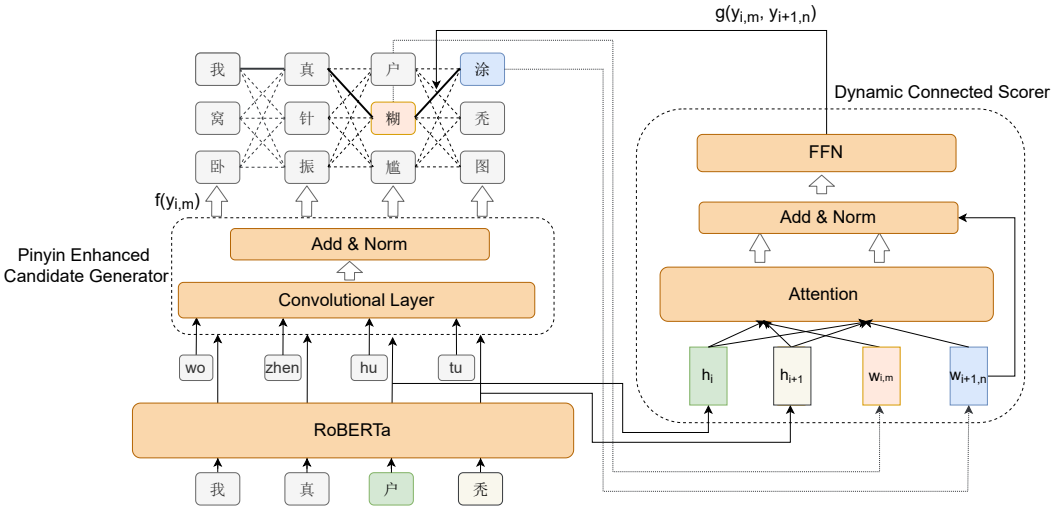


Figure 1: The architecture of DCN. Here, we only illustrate how to calculate one connection score between candidates "糊" and "涂" by the Dynamic Connected Scorer.

to automatically correct the incorrect part of the Chinese sentence and generate a correct target sequence $Y = \{y_1, y_2, ..., y_N\}$. Since the input sentence X and the output sentence Y have the same number of tokens (Chinese characters), pretrained non-autoregressive language models such as BERT are natural to be used in the CSC task. Given that non-autoregressive language models are based on the assumption of output independence, they will mismatch output Chinese characters and lead to the incoherent problem. This problem has also been mentioned in non-autoregressive machine translation (Gu et al., 2018, 2019; Gu and Kong, 2020) and pretrained language model (Yang et al., 2019).

3.2 Dynamic Connected Networks

To solve the above incoherence problem, we propose a novel model named Dynamic Connected Networks (DCN), which can learn the dependencies between output Chinese characters and alleviate the incoherence problem.

The model structure is illustrated in Figure 1. We use the RoBERTa (Liu et al., 2019; Cui et al., 2019) model as our base model. Firstly, RoBERTa with a Pinyin Enhanced Candidate Generator generates a series of candidate characters, and we sample k characters as candidates (the candidate generation method will be discussed in detail in the next subsection). For each two adjacent candidate characters, we learn the connection scores to determine the strength of the dependency between them by a dynamic connected scorer (DCScorer). The final correction score will be calculated by the joint prediction of connection scores and the prediction scores of the candidate generator at each position.

The DCScorer needs to consider the context information, the candidate characters of the current and next position simultaneously. Thus, we use the attention mechanism to learn the current candidate context representation p and next candidate context representation q. The strength of the dependency between two adjacent candidates is usually more related to the RoBERTa hidden representation of the current and next position, so the key and value in the attention mechanism contain only these two hidden representations. The DCScorer is formally defined as follows:

$$
\begin{aligned}
p_{i,m} &= \text{Attention}(Q_{i,m}W^Q, K_iW^K, V_iW^V) \\
q_{i,n} &= \text{Attention}(Q_{i+1,n}W^Q, K_iW^K, V_iW^V) \\
K_i = V_i &= \begin{bmatrix} h_i \\ h_{i+1} \end{bmatrix} \\
Q_{i,m} &= w_{i,m} \\
Q_{i+1,n} &= w_{i+1,n}
\end{aligned} \tag{1}
$$

where i is the character position, m and n are the indices of candidates of current position and next position respectively. Attention denotes the attention mechanism, where the Q, K, V denote query, key and value, and W denote the parameters to be learned in the attention layer. h is the hidden representation of the last transformer block, w denotes the candidate token embedding.

We add the candidate token embedding to the candidate context representation. Then we feed

the output into layer normalization and get two representations $p'_{i,m}$ and $q'_{i,n}$.

$$\begin{aligned} p'_{i,m} &= \text{LayerNorm}(p_{i,m} + w_{i,m}) \\ q'_{i,n} &= \text{LayerNorm}(q_{i,n} + w_{i+1,n}) \end{aligned} \tag{2}$$

We concatenate the two vectors and feed them into a feed-forward network (FFN) layer used by (Vaswani et al., 2017). Then we use a linear layer to calculate the connection score between the two candidates.

$$\begin{aligned} s &= \text{FFN}(\text{Concat}(p'_{i,m}, q'_{i,n})) \\ g(y_{i,m}, y_{i+1,n}) &= sv \end{aligned} \tag{3}$$

where v is a trainable weight vector and $g(y_{i,m}, y_{i+1,n})$ is the connection score between the mth candidate of ith position and nth candidate of $i+1$th position.

Since we feed k^2 pairs of candidate combinations into DCScorer, we will generate k^2 scores at each position. Eventually, the model will generate k^n candidate paths, and the score of each path is calculated using the following equations:

$$\text{S}(X, Y) = \sum_{i=1}^{N} f(y_{i,m}) + \sum_{i=1}^{N-1} g(y_{i,m}, y_{i+1,n}) \tag{4}$$

where y is the candidate character, $f(y_{i,m})$ is the prediction score of Pinyin Enhanced Candidate Generator for mth candidate of ith position.

3.3 Candidate Generation

We generate the candidate Chinese characters via a Pinyin Enhanced Candidate Generator based on RoBERTa.

Pinyin Enhanced Candidate Generator According to statistics, more than 80% spelling errors are related to phonological similarity (Liu et al., 2010). Since phonological errors account for a large proportion of Chinese character errors, a suitable method of introducing phonological information would be of great help in generating the candidates and correcting spelling errors.

The conversion from a single Chinese Pinyin to the Chinese character has a large ambiguity. It is difficult to convert properly because one Pinyin usually corresponds to many Chinese characters. However, when there are multiple consecutive Pinyin, we will have more confidence to convert Pinyin into correct Chinese characters. For example, the Pinyin of Chinese characters "户" and "糊" is "hu", and the Pinyin of "秃" and "涂" is "tu". When "hu" and "tu" are together, it will have a high probability of being converted to "糊涂" which means "confused" in Chinese. This is also a basic assumption used in Chinese Pinyin input methods.

Based on this, we propose a Pinyin Enhanced Candidate Generator, which can effectively reduce the ambiguity and generate better Chinese characters. The architecture is shown in Figure 1. Concretely, we adopt a convolutional layer to encode consecutive Pinyin and add the output of convolutional layer, hidden representation of RoBERTa and character embedding together. Then we feed the sum to layer normalization and get the prediction score $f(y_{i,m})$ via a linear layer. The equations are as follows:

$$\begin{aligned} c_i &= \text{Conv}(p''_{i-1}, p''_i, p''_{i+1}) \\ o_i &= \text{LayerNorm}(c_i + w_i + h_i) \\ f(y_{i,m}) &= o_i v'_m \end{aligned} \tag{5}$$

where p'' is the Pinyin embedding, w_i is the Chinese character embedding, h_i is the last hidden representation of RoBERTa, v'_m is the trainable weight vector for mth candidate.

There are various ways to represent Pinyin, and we find that simply representing each Pinyin without tone as a separate embedding can achieve good performance. We also try to encode Pinyin by Multi-Layer Perceptron (MLP) and GRU (Chung et al., 2014) encoder, which treat each letter of Pinyin as an embedding vector. Since they cannot achieve better results, we simply represent each Pinyin as a separate embedding in our following experiments.

Candidate Sampling Method Given the large number of usual Chinese characters, we sample the candidate characters for learning. We try several sampling methods and find that selecting the characters with the top-k prediction scores from vocabulary performs best. This also shows that the more difficult candidates can be used as negative training examples to effectively improve the discriminatory ability of the model. Therefore, all the main experimental results are based on the top-k sampling.

3.4 Learning

Loss function The probability for the sequence Y can be approximated by the following equation

$$p(Y|X) = \frac{e^{\text{S}(X,Y)}}{\sum_{Y'_i} e^{\text{S}(X,Y'_i)}} \tag{6}$$

Training Sets	# Line	Avg. Length	# Erroneous Sent.
Wikipedia	7,756,725	47.0	-
(Wang et al., 2018)	271,329	44.4	271,329
SIGHAN 2013	700	49.2	350
SIGHAN 2014	3,435	49.7	3,432
SIGHAN 2015	2,339	30.0	2,339
Test Sets	**# Line**	**Avg. Length**	**# Erroneous Sent.**
SIGHAN 2013	1,000	74.1	996
SIGHAN 2014	1,062	50.1	529
SIGHAN 2015	1,100	30.5	550

Table 2: Statistics of datasets.

where Y_i' are the path generated by the candidate characters.

The loss function is the maximum likelihood of the probability distribution denoted as

$$Loss = \begin{cases} -\log(p(Y|X)) & \mathrm{S}(X,Y) < \mathrm{S}_{max}(X,Y') \\ 0 & \mathrm{S}(X,Y) \geq \mathrm{S}_{max}(X,Y') \end{cases} \quad (7)$$

The loss function is similar to the one used by LSTM-CRF (Huang et al., 2015). It learns only the sampled negative candidate characters and the dependencies between them, which will unduly degrade the ranking of potential candidates. This possibly makes more similar candidates have a lower ranking. In order to avoid the above problem, we make a restriction on the loss function by setting its loss to 0 when the gold score is higher than or equal to the max score of all the candidate paths.

Pretraining The dependencies between Chinese characters can be more sufficiently learned via a large scale training corpus. In this paper, we pretrain our proposed model using Chinese Wikipedia data shown in Table 2. We randomly replace 15% of the characters, including 70% MASK, 15% characters from the confusion set, and 15% random characters. We exploit the confusion set released from SIGHAN 2013 (Wu et al., 2013) which consist of pronunciation similarity and shape similarity characters. Based on the RoBERTa model, we freeze the main parameters and only fine-tune the Pinyin Enhanced Candidate Generator and the Dynamic Connected Scorer.

3.5 Predicting

In the predicting stage, the top-k candidate characters from vocabulary are generated by the Pinyin Enhanced Candidate Generator. Eventually, there are k^n paths. In order to quickly select the path with the highest score, we use the Viterbi algorithm (Rabiner, 1989) based on dynamic programming to decode the output sequence.

4 Experiments

4.1 Experimental Setup

Datasets We use the large automatically generated corpus (Wang et al., 2018)[1] as our training data. In addition, the training sets of SIGHAN 2013, SIGHAN 2014, and SIGHAN 2015 are also included. For the pre-training method, we use the Chinese Wikipedia texts which have been converted to simplified Chinese characters.

We evaluate our proposed model on the test sets from SIGHAN 2013, SIGHAN 2014, and SIGHAN 2015 benchmarks. Similar to the previous works, we convert the traditional characters to simplified characters by OpenCC[2].

In order to evaluate our model more reasonably, we take 500 sentences from the SIGHAN training sets and the corresponding corrected results of these 500 sentences together as the validation set. The statistic information of all the datasets is listed in Table 2.

Evaluation Metrics To compare with the state-of-the-art models, We use the widely adopted sentence-level precision, recall, and F1-score as our evaluation method, which has been used by Hong et al. (2019)[3] and Cheng et al. (2020).

Baseline Models We compare our model with several state-of-the-art models.

- FASPell (Hong et al., 2019): This model uses the phonological and visual similarity information to select candidate characters.
- Soft-Masked BERT (Zhang et al., 2020): This method combines a detection network and a correction network based on BERT.
- SpellGCN (Cheng et al., 2020): This model incorporates phonological and visual similarity knowledge into BERT via a specialized graph convolutional network.
- Chunk-based method (Bao et al., 2020): This method utilizes a chunk-based framework and extends the traditional confusion sets with semantical candidates to cover different types of errors.

Model Hyperparameters We use RoBERTa-wwm (Cui et al., 2019) as our base model in this

[1]https://github.com/wdimmy/Automatic-Corpus-Generation

[2]https://github.com/BYVoid/

[3]https://github.com/iqiyi/FASPell

Dataset	Model	Detection-level			Correction-level		
		D-P	D-R	D-F	C-P	C-R	C-F
CSC13	FASPell (Hong et al., 2019)	76.2	63.2	69.1	73.1	60.5	66.2
	BERT (Cheng et al., 2020)	79.0	72.8	75.8	77.7	71.6	74.6
	SpellGCN (Cheng et al., 2020)	80.1	74.4	77.2	78.3	72.7	75.4
	SpellGCN*	85.2	77.7	81.2	83.4	76.1	79.6
	RoBERTa (Ours)	85.4	77.7	81.3	83.9	76.4	79.9
	RoBERTa-DCN (Ours)	86.2	78.4	82.1	84.6	76.9	80.5
	RoBERTa-Pretrain-DCN (Ours)	**86.8**	**79.6**	**83.0**	**84.7**	**77.7**	**81.0**
CSC14	FASPell (Hong et al., 2019)	61.0	53.5	57.0	59.4	52.0	55.4
	BERT (Cheng et al., 2020)	65.6	68.1	66.8	63.1	65.5	64.3
	SpellGCN (Cheng et al., 2020)	65.1	69.5	67.2	63.1	67.2	65.3
	RoBERTa (Ours)	64.2	68.4	66.2	62.7	66.7	64.6
	RoBERTa-DCN (Ours)	**67.6**	68.6	68.0	64.9	65.9	65.4
	RoBERTa-Pretrain-DCN (Ours)	67.4	**70.4**	**68.9**	**65.8**	**68.7**	**67.2**
CSC15	FASPell (Hong et al., 2019)	67.6	60.0	63.5	66.6	59.1	62.6
	Soft-Masked BERT (Zhang et al., 2020)	73.7	73.2	73.5	66.7	66.2	66.4
	BERT (Cheng et al., 2020)	73.7	78.2	75.9	70.9	75.2	73.0
	SpellGCN (Cheng et al., 2020)	74.8	80.7	77.7	72.1	77.7	*75.9*(74.8)
	RoBERTa (Ours)	74.7	77.3	76.0	72.1	74.5	73.3
	RoBERTa-DCN (Ours)	76.6	79.8	78.2	74.2	77.3	75.7
	RoBERTa-Pretrain-DCN (Ours)	**77.1**	**80.9**	**79.0**	**74.5**	**78.2**	**76.3**

Table 3: Experimental results of sentence-level precision, recall, and F1-score (%). D, C denote the detection and correction respectively. Since "的", "地", "得" are rarely distinguished on SIGHAN 2013, we remove all the related correction results. To compare more fairly with SpellGCN, we rerun the released code of Cheng et al. (2020) and remove all the related correction results. The results are reported with SpellGCN*. The reported result of SpellGCN on SIGHAN 2015 is not correct, where the precision, recall and F-score don't match. If the precision and recall are correct, F-score should be 74.8.

paper. We utilize AdamW (Loshchilov and Hutter, 2019) optimizer with learning rate of 5e-5. The training batch size is set to 32, and we train 12 epochs for all the experiments. To better learn the dependencies between characters, we learn the DCN model with MASK token for the first 2 epochs the same with the pretraining method. The number of candidates k for training is set to 5 and the number for predicting is set to 8. The convolution window size of the Pinyin Enhanced Candidate Generator is set to 3. The dimensions of all the hidden representations are 768. We search learning rate from {2e-5, 3e-5, 5e-5} and select the best model on the validation set.

4.2 Experimental Results

The experimental results are shown in Table 3. Our proposed RoBERTa-DCN model has the best detection and correction performance on the three SIGHAN test sets. Both FASPell and SpellGCN models use sophisticated techniques to incorporate the phonological and visual information and achieve a relatively good performance. Our DCN model is more focused on the incoherence problem and modeling the dependencies of the output tokens. Our proposed model exceeds FASPell and SpellGCN by simply using a Pinyin Enhanced Candidate Generator to model the phonological information, which also illustrates the effectiveness of DCN.

When we pre-train DCN using wiki data, the model gets further improvement in the effectiveness. This indicates that modeling the dependencies between output Chinese characters is important. DCN may achieve better performance if more data are used to learn the dependencies.

Soft-Masked BERT uses detection network and correction network simultaneously. In contrast, our DCN model predicts the target sequence directly, and the different tokens between the input sequence and the target sequence are regarded as the detection results. As shown in the experimental results, compared to Soft-Masked BERT, our method improves 5.5% and 9.9% on detection and correction respectively.

To compare with some other state-of-the-art works, we also evaluate our proposed model using the official evaluation toolkit[4] of SIGHAN 2015 in Table 4. The Chunk-based method, which uses a series of methods to construct the candidate set,

[4]http://nlp.ee.ncu.edu.tw/resource/csc.html

Model	Detection-level				Correction-level			
	D-Acc	D-P	D-R	D-F	C-Acc	C-P	C-R	C-F
Chunk-based method (Bao et al., 2020)	76.8	88.1	62.0	72.8	74.6	87.3	57.6	69.4
BERT (Cheng et al., 2020)	83.0	85.9	78.9	82.3	81.5	85.5	75.8	80.5
SpellGCN (Cheng et al., 2020)	83.7	85.9	80.6	83.1	82.2	85.4	77.6	81.3
RoBERTa (Ours)	83.2	86.6	78.6	82.4	81.8	86.2	75.8	80.7
RoBERTa-DCN (Ours)	84.2	86.4	**81.1**	83.7	82.8	86.0	**78.4**	82.0
RoBERTa-Pretrain-DCN (Ours)	**84.6**	**88.0**	80.2	**83.9**	**83.2**	**87.6**	77.3	**82.1**

Table 4: The performance evaluated by official tools on SIGHAN 2015.

Sampling Method	D-F	C-F
Top-k of vocabulary	**89.7**	**88.7**
Multinomial distribution sampling	88.1	87.6
Random sampling	12.2	7.3
Top-k of confusion set	35.8	34.7

Table 5: Effect of the candidate generation methods.

Model	D-P	D-R	D-F	C-P	C-R	C-F
RoBERTa-DCN	**89.8**	**89.6**	**89.7**	**88.8**	**88.6**	**88.7**
- PECGenerator	87.7	88.4	88.1	86.7	87.4	87.1
- DCScorer	87.4	88.4	87.9	86.8	87.8	87.3
- weighted loss	87.6	89.0	88.3	86.8	88.2	87.5
RoBERTa	86.1	89.2	87.6	85.5	88.6	87.0

Table 6: Ablation Study of DCN on validation set. PECGenerator is the Pinyin Enhanced Candidate Generator. Weighted loss refers to the condition of loss. When we remove the PECGenerator, the RoBERTa generates the candidates by predicting the candidate characters. When the DCScorer is removed, the model selects the top-1 predicted result as the correct character.

achieves good performance for precision. However, the recall of this method is relatively low, and the F-score of our method significantly outperforms the chunk-based method by more than 10%. Similarly, our model also achieves a better result than SpellGCN.

4.3 Effect of Candidate Generation

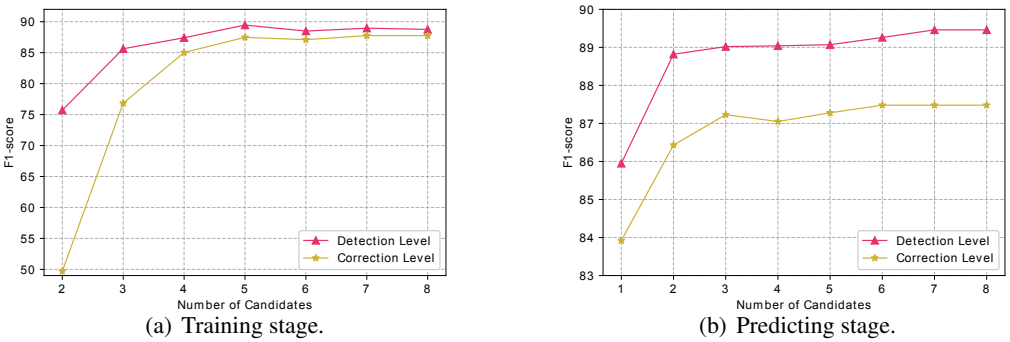
The performance of DCN varies with the candidate generation strategy and the number of sampled candidate characters. We compare the effects of four sampling methods for training, which are sampling top-k candidates from vocabulary, sampling top-k candidates from the confusion set, random sampling from the vocabulary and sampling from a multinomial distribution. For the multinomial distribution sampling, the probabilities are obtained from the Softmax output of the Pinyin Enhanced Candidate Generator. All the subsequent experiments are conducted on the validation set. The experimental results are shown in Table 5.

From Table 5, we can see that the top-k of vocabulary method has the best performance. The multinomial distribution sampling also has a good performance, while the random sampling and top-k of confusion set cannot achieve good performance. This means that sampling some difficult candidates is more beneficial to the model training to improve the model discriminative ability.

We also conduct experiments with the effect of the number of candidates. Figure 2(a) shows the change curve of the effect when the number of candidates for training increase. The effect gradually gets better as the number of candidates increases at the beginning, and the effect no longer has a significant improvement after the number of candidates for prediction exceeds 5. Figure 2(b) shows the performance when we fix the number of training candidates as 5 and increase the number of prediction candidates. The performance keeps improving as the number of prediction candidates increases.

4.4 Ablation Study

We conduct a series of experiments to determine which component in the DCN model plays a more important role. Table 6 shows the results of our experiments. When we remove the Pinyin Enhanced Candidate Generator, both the detection and correction F-scores decrease about 1.5%. This demonstrates that the phonological information plays an important role in candidates generation methods. When we remove the dynamic connected scorer, the detection F-score decreases nearly 2%, which indicates that the dependencies between Chinese characters are important for the CSC task. Similarly, the weighted loss also help our models improve the performance.



(a) Training stage. (b) Predicting stage.

Figure 2: The effect of the number of candidates.

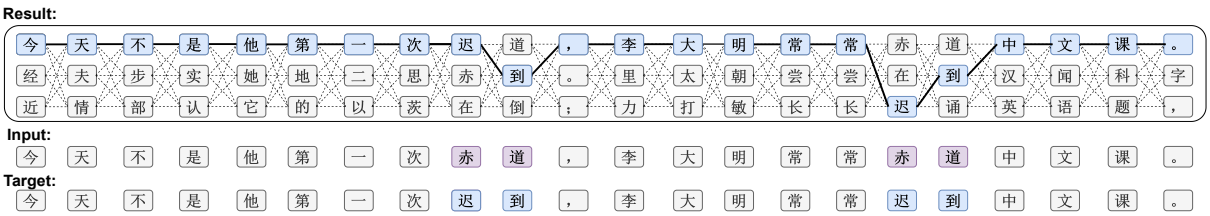


Figure 3: An example of Viterbi decoding. "赤道" in this sentence should be corrected to "迟到". Translation of this example: *Today is not the first time he is late. Daming Li is often late to Chinese class.*

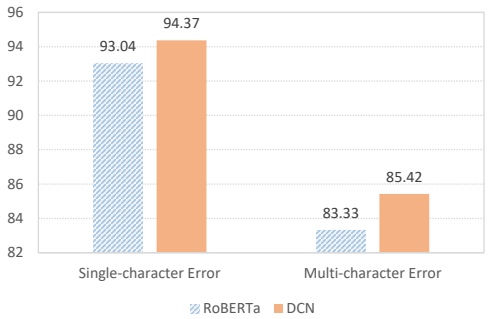


Figure 4: The comparison of detection F-score between RoBERTa and DCN on single-character error and multi-character error validation sets.

4.5 Case Study and Analysis

We find that the DCN model performs better than the vanilla RoBERTa model on consecutive errors. Figure 3 shows an example of consecutive errors. The vanilla RoBERTa model cannot detect it well and can only partially correct it because consecutive errors will influence each other. In contrast, DCN can correct it completely. The best path for this example is shown in this figure. The correct Chinese characters "迟到" did not rank first, but the path including "迟到" have the highest score because they are more fluent than other candidate combinations. This example also shows that our model can alleviate the incoherent problem.

Figure 4 shows the detection F-score of RoBERTa and DCN on single-character and multi-character error sets. DCN performs better than RoBERTa in both single-character and multi-character level cases. The effect of DCN is more obvious in the multi-character cases, which also shows that DCN has some advantages for multi-character type errors. At the same time, the performance of single-character error cases is much better than multi-character error cases, which indicates that DCN still has much room to improve for multi-character errors.

By comparing the results of RoBERTa and RoBERTa with the candidate generator. We find that 96.7% of the correct characters are in the top-5 candidates of RoBERTa. In contrast, 98.6% of the correct characters are in the top-5 candidates of RoBERTa with candidate generator. This result illustrates that the Pinyin Enhanced Candidate Generator can generate better candidates.

5 Conclusion

In this paper, we propose a novel model named DCN to solve the incoherent problem in the CSC task. To better incorporate the phonological information, we propose a simple and effective Pinyin Enhanced Candidate Generator. The experimental

results show that our proposed model has achieved the state-of-the-art performance on three datasets. DCN may also be utilized on other tasks such as non-autoregressive machine translation. As for future work, how to make better use of phonological and visual information still needs to be discussed.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (No. 2018YFC0832302). I would like to thank the anonymous reviewers and Xiaoxue Wang for their insightful comments and suggestions.

References

Zuyi Bao, Chen Li, and Rui Wang. 2020. Chunk-based Chinese spelling check with global optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2031–2040, Online. 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Jiatao Gu and Xiang Kong. 2020. Fully non-autoregressive neural machine translation: Tricks of the trade. *arXiv preprint arXiv:2012.15833*.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

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 *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 160–169, Hong Kong, China. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

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 2010: Posters*, pages 739–747, Beijing, China. Coling 2010 Organizing Committee.

Xiaodong Liu, Kevin Cheng, Yanyan Luo, Kevin Duh, and Yuji Matsumoto. 2013. A hybrid Chinese spelling correction using language model and statistical machine translation with reranking. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 54–58, Nagoya, Japan. Asian Federation of Natural Language Processing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. *arXiv preprint arXiv:1909.01187*.

Lawrence R Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for Chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527, Brussels, Belgium. Association for Computational Linguistics.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for Chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at SIGHAN bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42, Nagoya, Japan. Asian Federation of Natural Language Processing.

Yang Xin, Hai Zhao, Yuzhu Wang, and Zhongye Jia. 2014. An improved graph model for Chinese spell checking. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 157–166, Wuhan, China. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

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