Modalities Should be Appropriately Leveraged: Uncertainty Guidance for Multimodal Chinese Spelling Correction

Yongliang Lin[1], **Zhen Zhang**[1], **Mengting Hu**[1], **Yufei Sun**[1], **Yuzhi Zhang**[1,2*]
[1]College of Software, Nankai University
[2]Haihe Lab of ITAI
linyongliang232@hotmail.com, zhzhen23@gmail.com, mthu@mail.nankai.edu.cn
yufei_sun@sina.com, zyz@nankai.edu.cn

Abstract

Chinese spelling correction (CSC) aims to detect and correct spelling errors in Chinese texts. Most spelling errors are phonetically or graphically similar to the correct ones. Thus, recent works introduce multimodal features to obtain achievements. In this paper, we found that different spelling errors have various biases to each modality, highlighting the importance of appropriately exploiting multimodal features. To achieve this goal, we propose the UGMSC framework, which incorporates uncertainty into both the feature learning and correction stages. Specifically, the UGMSC framework makes predictions with multimodal features and estimates the uncertainty of the corresponding modalities. Then it dynamically fuses the features of all modalities for model learning, and performs spelling correction under the uncertainty-guided strategy. Experimental results on three public datasets demonstrate that the proposed approach provides a significant improvement compared with previous strong multimodal models. The proposed framework is model-agnostic and can be easily applied to other multimodal models.

Keywords: Chinese spell correction, multimodal, uncertainty guided

1. Introduction

Chinese Spelling Correction (CSC) is an important and fundamental task in Chinese NLP research, which aims to detect and correct spelling mistakes in Chinese texts. Unlike English, Chinese has many pictographic and polysyllabic characters, and there are no word separators between Chinese characters. Chinese spelling errors are constantly present in human writing, automatic speech recognition (ASR), and optical character recognition (OCR) systems (Yu et al., 2014), making CSC a critical task for many language applications. According to Liu et al. (2010), about 83% of spelling errors are phonologically similar to the correct ones, and 48% of errors are graphically similar to the correct ones.

Early works have been done on the CSC task with generative language model (Liu et al., 2013; Yu and Li, 2014), heuristic methods (Chang et al., 2015; Chu and Lin, 2015), sequence-to-sequence models (Wang et al., 2019), with the rise of pre-trained language models such as BERT (Devlin et al., 2019), researches focus on pre-trained models (Hong et al., 2019; Zhang et al., 2020; Cheng et al., 2020). Recently, multimodal methods have received great attention from the academic community (Cheng et al., 2020; Zhang et al., 2021; Xu et al., 2021; Huang et al., 2021; Liu et al., 2021), suggesting that multimodal information is really helpful for CSC task.

Although multimodal information is beneficial, we found that different spelling errors have various biases in each modality. Table 1 shows Chinese misspellings in phonetically similar and graphically similar cases. In the first case, we can correct the misspelled character "高" with a phonetically similar candidate: "告". In the second case, we can easily find the correct character "饱" in graphically similar candidates. Except for the above two modalities, the corresponding Hanyu Pinyin[1] (pinyin) is also a crucial feature. The sequence of the Chinese word "很跑" is "hěn, pǎo", which is not a valid pinyin sequence, so models can find more position evidence of the misspelling.

Phonetically Similar Case	
Input	我以前想要高 (gāo)诉你。
PSC	告 (gào)　皋 (gāo)　稿 (gǎo)　膏 (gāo)
GSC	富 (fù)　镐 (gǎo)　亮 (liàng)　嵩 (sōng)
Correct	我以前想要告 (gào)诉你。
Translate	I wanted to tell you before.
Graphically Similar Case	
Input	他睡很跑 (pǎo)，睡到忘了时间起床。
PSC	炮 (pào)　抛 (pāo)　袍 (páo)　好 (hǎo)
GSC	包 (bāo)　饱 (bǎo)　抱 (bào)　泡 (pào)
Correct	他睡很饱 (bǎo)，睡到忘了时间起床。
Translate	He slept so well that he forgot to get up at the right time.

Table 1: Examples of Chinese misspellings and phonetically similar candidates (PSC), graphically similar candidates (GSC) of the misspelled characters. The misspellings and the corresponding pinyins are highlighted in red, the correct ones are in blue.

[*] Yuzhi Zhang is the corresponding author.

[1] Pinyin is the official phonetic system of Chinese

LREC-COLING 2024, pages 11463–11474

The above cases suggest that modalities play various importance when correcting different spelling errors. Therefore, **models need to learn how to determine which modality is important, and how to deal with multimodal information appropriately.** Motivated by this, we propose UGMSC[2] (Uncertainty Guided Multimodal Spelling Correction), which is a framework that leverages multimodal information under uncertainty guidance. The guidance is integrated into both the feature learning and correction phases. The rationality relies on that uncertainty reflects the importance of the modality information and the reliability that it can be utilized for making spelling correction decisions.

Specifically, during the feature learning phase, we build modality correctors, including three partial correctors and a joint corrector, to make modality-specific decisions. These decisions are exploited for estimating uncertainties, representing the modality's trustworthiness and importance. They guide the learning and weighting of features. Then, in the correction phase, we develop an uncertainty-aware correction module to further guide the multimodal knowledge. This module fuses the modality-specific decisions to yield the final correction decision.

The contributions of this paper can be summarized as follows:

- To the best of our knowledge, we are the first to investigate the importance of appropriately using multimodal information in CSC task.
- We propose UGMSC, which is a framework to leverage multimodal information in an appropriate manner under the guidance of modality uncertainty. We design modality correctors and uncertain-aware correction modules to incorporate uncertainty into feature and decision steps, respectively.
- Extensive experiment results show that our method achieves great progress compared to strong multimodal baseline methods. Our framework is also model-agnostic, indicating that it can be easily applied to various approaches. The results demonstrate the effectiveness and validity of our method.

2. Related Work

2.1. Chinese Spelling Correction

Manual rule based methods (Chu and Lin, 2015; Jiang et al., 2012) are proposed in the early era of CSC tasks. Then, traditional machine learning algorithms like Conditional Random Field (Wang and Liao, 2015) and Hidden Markov Model (Zhang et al., 2015) and statistics methods like Support Vector Machine (Chen et al., 2013; Yu and Li, 2014; Liu et al., 2013) are used in researches of this area, these methods focus on detecting the errors and making decisions on the candidates of the errors. Afterward, neural network based methods rose and became the mainstream of the CSC task. Han et al. (2019); Wang et al. (2018) apply Bi-LSTM to the CSC task regarding it as a sequence labeling problem. Besides, the confusion set of Chinese characters is widely used for handling the similarity among the characters better (Wang et al., 2019). Thanks to the great breakthrough of NLP made by Transformer (Vaswani et al., 2017), pre-trained language models such as BERT (Devlin et al., 2019) are brought into the CSC task and become the backbone of many recent works (Hong et al., 2019; Zhang et al., 2020; Wang et al., 2021; Zhu et al., 2022).

2.2. Multimodal Chinese Spelling Correction

Multimodal methods are also generally favored in the CSC task due to the regular pattern of Chinese misspellings. SpellGCN (Cheng et al., 2020) builds two similarity graphs for phonetic and graphic similarities and makes use of them by graph convolution network. PHMOSpell (Huang et al., 2021) applies an adaptive gating module on phonetic and graphic modal representation vectors to compute prelogits vectors. ReaLiSe (Xu et al., 2021) develops modality feature encoders and a selective modality fusion mechanism to obtain a multimodal representation vector for prediction. PLOME (Liu et al., 2021) and CoSPA (Yang and Yu, 2022) fuse multimodal representation vectors and feed them into a Transformer encoder by simply summing them and word embedding vectors. Different from them, we argue that leveraging multimodal features appropriately is also crucial. We propose to utilize every modality under the guidance of uncertainty.

3. Methodology

3.1. Problem Formulation and Overview

Given an original sequence consisting of N Chinese characters $\mathbf{X} = (x_1, x_2, \cdots, x_N)$, which contains some error characters, the objective of the CSC task is to output the correct sequence $\mathbf{Y} = (y_1, y_2, \cdots, y_N)$, where $\mathbf{X}$ and $\mathbf{Y}$ have the same length. In most cases, mistaken characters are only in the minority of the input X.

As shown in Figure 1, the proposed UGMSC is a framework consisting of three main modules,

[2]Our code are available at https://github.com/LYL232/UGMSC

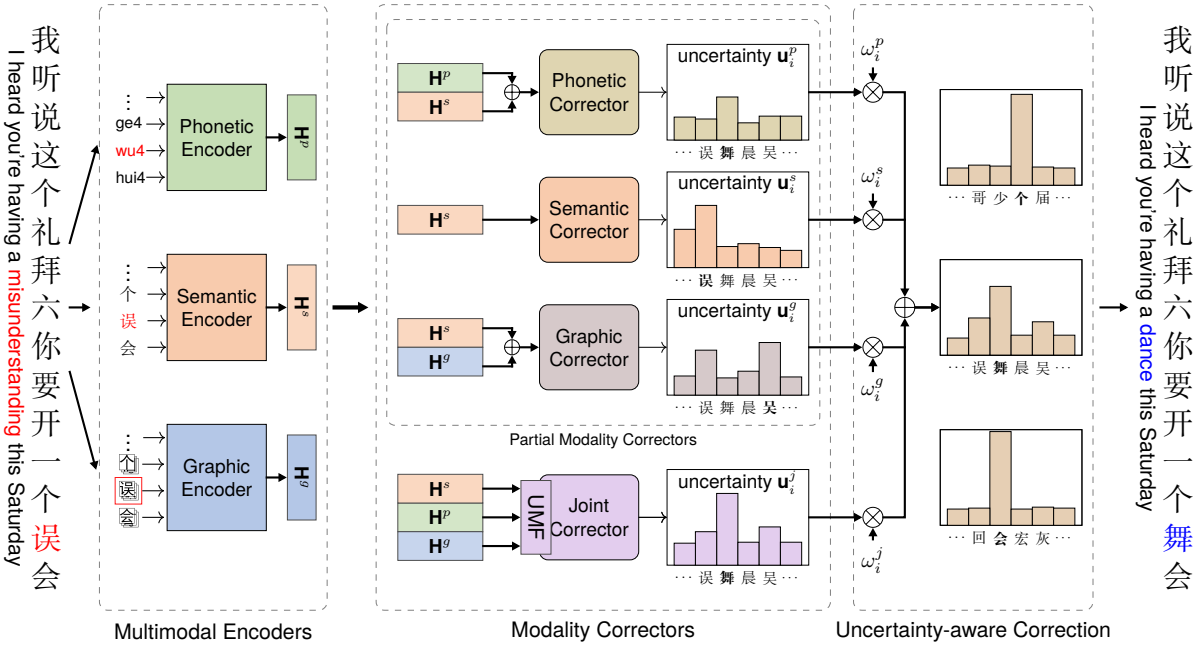

Figure 1: The architecture of UGMSC. Multimodal encoders provide the representation vectors of the modalities. Modality correctors learn the representation vectors in different ways to provide correction predictions and estimate the uncertainty of the corresponding modalities. The joint corrector leverages all the representation vectors through the UMF mechanism. Then an uncertainty-aware correction module is applied to make decisions among the predictions of all these correctors.

including multimodal encoders, modality correctors, and the uncertainty-aware correction module. First, the input sequence $\mathbf{X}$ is encoded by multimodal encoders to extract modality features. Then, the encoded representation vectors are fed to modality correctors, where each modality corrector predicts the correct token with different modality features and estimates the uncertainty of the prediction. A special modality corrector, called the joint corrector, learns the features of all modalities fused by the UMF(Uncertainty-aware Modality Fusion, introduced in §3.3.2) mechanism, which introduces uncertainty for better modalities feature learning. The uncertainty-aware correction module then makes decisions among all the predictions based on all the uncertainties estimated by all modality correctors.

3.2. Multimodal Encoders

Each encoder extracts a representation vector of a single modality for each input Chinese character. In this paper, we empirically use three encoders to extract features from semantic, phonetic, and graphic modalities, respectively.

Semantic Encoder. The semantic encoder extracts the representation vectors of the semantic modality through word embeddings, it first projects $\mathbf{X}$ to word embedding vectors, and does further processing like sequence learning with Transformer blocks over the embedding vectors to get the representation vectors of the semantic modality: $\mathbf{H}^s = (\mathbf{h}_1^s, \mathbf{h}_2^s, \cdots, \mathbf{h}_N^s)$.

Phonetic Encoder. The phonetic encoder learns the pinyin or acoustic features of the input characters and outputs the representation vectors of the phonetic modality: $\mathbf{H}^p = (\mathbf{h}_1^p, \mathbf{h}_2^p, \cdots, \mathbf{h}_N^p)$. One practicable approach is to consider pinyin as another form of character embedding and apply sequence learning over these embeddings.

Graphic Encoder. The graphic encoder encodes the graphic features, such as the images of the input characters $\mathbf{X}$, into the representation vectors of the graphic modality: $\mathbf{H}^g = (\mathbf{h}_1^g, \mathbf{h}_2^g, \cdots, \mathbf{h}_N^g)$. In general, convolutional neural networks (CNNs) with character image input are helpful.

3.3. Modality Correctors

As described in Table 1, error tokens have various biases towards each modality. The naive modality features $\mathbf{H}^s$, $\mathbf{H}^p$, and $\mathbf{H}^g$ should be calibrated. Modality correctors aim to predict the correct token sequence $\mathbf{Y}$ based on the representation vectors of particular modalities and to estimate the uncer-

tainty of their predictions, which indicates the reliability of the corresponding modalities.

Structure. All the correctors, including three partial modality correctors (the phonetic, semantic, and graphic correctors) and one joint correct, have a similar structure: Transformer blocks and a shared linear mapping. A corrector's prediction procedure can be formulated as follows:

$$\begin{aligned} \mathbf{R} &= \text{Transformer}_l(\mathbf{H}_0)\mathbf{W} + \mathbf{b}, \\ \mathbf{P} &= \text{Softmax}(\mathbf{R}), \end{aligned} \tag{1}$$

where Transformer_l is a l-layer Transformer, $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N)$ is the predict logits. $\mathbf{W}, \mathbf{b}$ are the parameters of the shared linear mapping. $\mathbf{H}_0$ is the representation vector sequence fed into the corrector, where the details are introduced in §3.3.1 and §3.3.2. $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N)$ is the predicted probability distribution of the corrector.

Uncertainty Computation. Previous works are usually evaluated on two metrics: detection, indicating whether a token is correct in a binary manner; correction, aiming to put right the error tokens (Liu et al., 2021; Yang and Yu, 2022). Therefore, we design to compute uncertainties for both metrics that reflect how reliable the information is in the detection and correction tasks. Concretely, uncertainty is computed via the information entropy (Shannon, 1948) of the correction and detection distribution over the prediction. The detection distribution is a 0-1 distribution, where "1" means that the input token is misspelled, the possibility is $\mathbf{p}_i(y = x_i)$, i.e. the probability that the corrector considers the input token x_i to be correct. The correction distribution is the probability distribution of each token in the vocabulary, i.e. $\mathbf{p}_i$. Thus, the uncertainty metrics can be formalized as follows:

$$\begin{aligned} \mathcal{E}_d(\mathbf{p}_i) &= -\mathbf{p}_i(y = x_i)\log\mathbf{p}_i(y = x_i) \\ &\quad - \mathbf{p}_i(y \neq x_i)\log\mathbf{p}_i(y \neq x_i), \\ \mathcal{E}_c(\mathbf{p}_i) &= -\sum_{v \in V} \mathbf{p}_i(y = v)\log\mathbf{p}_i(y = v), \\ \mathbf{u}_i &= \left(\mathcal{E}_c(\mathbf{p}_i), \mathcal{E}_d(\mathbf{p}_i)\right), \end{aligned} \tag{2}$$

where $\mathcal{E}_d$ is the detection level entropy, $\mathcal{E}_c$ is the correction level entropy. V is the vocabulary set, $\mathbf{u}_i$ is the uncertainty metrics of the corrector for the token x_i.

3.3.1. Partial Modality Correctors

Partial modality correctors make predictions based on the information of partial modalities, these correctors should cover all modalities, and then we can infer the reliability of the modalities according to the uncertainty metrics given by all partial modality correctors. In this paper, we build three partial modality correctors: the semantic corrector makes predictions based on the semantic modality representation vectors, so the input features sequence of the semantic corrector is $\mathbf{H}_0^s = \mathbf{H}^s$. The phonetic corrector and the graphic corrector make predictions based on the phonetic and graphic modalities, respectively. Since there are so many polysyllabic and graphically similar characters, it is difficult to predict the exact character only based on the phonetic or graphic information alone. Thus, we add the semantic information as an augmentation, by simply summing $\mathbf{H}^s$ to the representation vectors of the corresponding modality, so the input features of the phonetic corrector and the graphic corrector can be computed as $\mathbf{H}_0^p = \mathbf{H}^p + \mathbf{H}^s$ and $\mathbf{H}_0^g = \mathbf{H}^g + \mathbf{H}^s$, respectively. We can also use a learnable linear mapping of the concatenation of $\mathbf{H}^s$ and the representation vectors of the corresponding modality to be the input features if $\mathbf{H}^s, \mathbf{H}^p, \mathbf{H}^g$ are of different sizes.

The augmented features are further fed into their own partial modality correctors to calculate the logits and probability distributions. We obtain the logits $\mathbf{R}^s$, $\mathbf{R}^p$, and $\mathbf{R}^g$ from the semantic, phonetic, and graphic correctors, respectively. Their probability distributions are denoted as $\mathbf{P}^s$, $\mathbf{P}^p$, and $\mathbf{P}^g$.

3.3.2. Joint Corrector

Uncertainty-aware Modality Fusion (UMF). The joint corrector leverages information from all modalities to make its predictions. For a better fusion of features from all modalities, we develop the uncertainty-aware modality fusion mechanism to fuse all the modality representation vectors under the guidance of the uncertainty metrics. We build a simple neural network composed of fully connected layers, i.e. multi-layer perceptron (MLP), to learn the features of the uncertainty metrics given by all partial correctors.

We observed that, in most cases, the prediction probability is very close to 1 or 0, so the uncertainty metrics are always close to 0 (see §4.7). Therefore, to better capture the relationship between all uncertainty metrics given by all partial correctors, we compute the log scale of these metrics, because it is easier for a MLP to distinguish log(0.01) and log(0.001) than 0.01 and 0.001. Formally, we normalize all uncertainty metrics to the interval (0,1) and compute their multi-scale features of them to feed the MLP:

$$\begin{aligned} \text{um}(\mathbf{u}) = \text{MLP}([\,&(u, 1-u, \log u, \log(1-u)) \\ &\text{for } u \in \mathbf{u}]), \end{aligned} \tag{3}$$

where $\mathbf{u}$ is a vector of normalized uncertainty metrics, $[\cdot]$ means the concatenation operation. the

MLP maps the extended features to the values in a specified size.

Then, the UMF mechanism fuses the representation vectors of all modalities under the guidance of the uncertainty metrics $\mathbf{u}_i^s, \mathbf{u}_i^p, \mathbf{u}_i^g$ estimated by partial modality correctors:

$$\begin{aligned} m^s, m^p, m^g &= \sigma\left(\text{um}([\mathbf{u}_i^s, \mathbf{u}_i^p, \mathbf{u}_i^g])\right), \\ \mathbf{h}_i^j &= \text{Fuse}(m^s \cdot \mathbf{h}_i^s, m^p \cdot \mathbf{h}_i^p, m^g \cdot \mathbf{h}_i^g), \end{aligned} \tag{4}$$

where m^s, m^p, m^g are positive scalars that indicate the reliability of the semantic, phonetic, graphic modality respectively. $\sigma(\cdot)$ is the Sigmoid function. Fuse is a simple operation that merges all weighted modality representation vectors into the input vector of the joint corrector, it can be a summation if all input vectors have the same size or a linear mapping of the concatenation of all input vectors.

Then, the joint corrector learns the fused modality representation vectors to make predictions:

$$\mathbf{H}_0^j = [\mathbf{h}_1^j, \mathbf{h}_2^j, \cdots, \mathbf{h}_N^j]. \tag{5}$$

The predict logits and the probability distribution of the joint corrector are denoted as $\mathbf{R}^j$ and $\mathbf{P}^j$.

3.4. Uncertainty-aware Correction

Although the joint corrector leverages information from all modalities, the predictions of partial correctors remain valuable, because the UMF mechanism is in some cases unable to filter out the confusing information from some modalities (see case 1 and case 2 in §4.7), so we make prediction decisions among the predictions of all correctors under the guidance of the uncertainty:

$$\begin{aligned} \omega_i^s, \omega_i^p, \omega_i^g, \omega_i^j &= \text{Softmax}\Big(\text{um}([\mathbf{u}_i^s, \mathbf{u}_i^p, \mathbf{u}_i^g, \mathbf{u}_i^j])\Big), \\ \mathbf{p}_i^u &= \text{Softmax}\left(\sum_{C\in\{s,p,g,j\}} \omega_i^C \mathbf{r}_i^C\right). \end{aligned} \tag{6}$$

Then we choose the index of $\mathbf{p}_i^u$ with the maximum probability to be the correction result for token x_i.

3.5. Direct Correctors Training

Since we make the prediction based on $\mathbf{p}_i^u$ in Eq.6, it is intuitive to optimize directly on $\mathbf{p}_i^u$, but this leads to unbalanced training: the joint corrector plays a major role, while the partial correctors perform ill and make insignificant contributions to the uncertainty-aware correction, e.g. in most cases, $\mathbf{p}_i^u \approx \mathbf{p}_i^j, w_i^j \approx 1, w_i^s \approx w_i^g \approx w_i^p \approx 10^{-6}$. We attribute this to the fact that the joint corrector has the strongest structure and it tackles full information from all modalities, which induces the uncertainty-correction correction module to increase the weight of the joint corrector predictions w_i^j in most cases and decrease the weights of the partial correctors w_i^s, w_i^g, w_i^p. Then the partial correctors receive more inaccurate gradients from backward, they are not well trained and unable to make accurate predictions, then w_i^s, w_i^g, w_i^p are optimized to become smaller, forming a vicious circle. So we devise direct correctors training as Eq. 7 to ensure that the partial correctors are properly optimized. We compute the cross entropy of all the prediction probability distributions given by all the correctors and $\mathbf{p}_i^u$, and then optimize the sum of them:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{\mathbf{p}\in\mathcal{P}} \log \mathbf{p}_i(y = y_i), \tag{7}$$

where $\mathcal{P} = \{\mathbf{p}^s, \mathbf{p}^p, \mathbf{p}^g, \mathbf{p}^j, \mathbf{p}^u\}$ is the set of all predict probability distributions.

4. Experiment

In this section, we apply our UGMSC framework to multimodal models SCOPE (Li et al., 2022) and ReaLiSe (Xu et al., 2021) for the CSC task and introduce the evaluation results on the SIGHAN benchmarks (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015). We then perform a model-agnostic experiment, ablation study, and case study to verify the effectiveness of UGMSC.

4.1. Dataset and Metrics

Dataset. Following the training settings of SCOPE, we use the datasets provided by ReaLiSe directly for training and evaluation. The training dataset consists of SIGHAN13 (Wu et al., 2013), SIGHAN14 (Yu et al., 2014), SIGHAN15 (Tseng et al., 2015) training data and the generated examples from (Wang et al., 2018), we evaluate our method on the test dataset of SIGHAN13, SIGHAN14, SIGHAN15. Following previous works (Wang et al., 2019; Cheng et al., 2020; Zhang et al., 2020), ReaLiSe converts the SIGHAN datasets to the Simplified Chinese using the OpenCC tool[3].

Metrics. We use sentence-level precision, recall, and F1 as our evaluation metrics which are widely used in the CSC task, including the detection and correction sub-tasks. In sentence-level metrics, a sentence is considered to be detected or corrected only if all the characters in the sentence are successfully detected or corrected.

[3] https://github.com/BYVoid/OpenCC

Dataset	Method	Detection Level				Correction Level			
		Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
SIGHAN13	FASpell (Hong et al., 2019)	-	76.2	63.2	69.1	-	73.1	60.5	66.2
	SpellGCN (Cheng et al., 2020)	-	80.1	74.7	77.2	-	78.3	72.7	75.4
	ReaLiSe† (Xu et al., 2021)	82.7	88.6	82.5	85.4	81.4	87.2	81.2	84.1
	UGMSC(ReaLiSe)†	82.9	89.0	82.9	85.9	82.3	**88.4**	82.3	85.2
	MLM-phonetics (Zhang et al., 2021)	-	82.0	78.3	80.1	-	79.5	77.0	78.2
	MDCSpell (Zhu et al., 2022)	-	**89.1**	78.3	83.4	-	87.5	76.8	81.8
	SCOPE† (Li et al., 2022)	-	87.4	**83.4**	85.4	-	86.3	82.4	84.3
	UGMSC(SCOPE)†	**83.6**	**89.1**	**83.4**	**86.2**	**82.9**	88.3	**82.7**	**85.4**
SIGHAN14	FASpell (Hong et al., 2019)	-	61.0	53.5	57.0	-	59.4	52.0	55.4
	SpellGCN (Cheng et al., 2020)	-	65.1	69.5	67.2	-	63.1	67.2	65.3
	ReaLiSe (Xu et al., 2021)	78.4	67.8	71.5	69.6	77.7	66.3	70.0	68.1
	UGMSC(ReaLiSe)	78.8	67.9	72.5	70.1	78.2	66.8	71.3	69.0
	MLM-phonetics (Zhang et al., 2021)	-	66.2	**73.8**	69.8	-	64.2	**73.8**	68.7
	MDCSpell (Zhu et al., 2022)	-	70.2	68.8	69.5	-	69.0	67.7	68.3
	SCOPE (Li et al., 2022)	-	70.1	73.1	71.6	-	68.6	71.5	70.1
	UGMSC(SCOPE)	**80.5**	**72.5**	71.2	**71.8**	**80.0**	**71.6**	70.2	**70.9**
SIGHAN15	FASpell (Hong et al., 2019)	-	67.6	60.0	63.5	-	66.6	59.1	62.6
	SpellGCN (Cheng et al., 2020)	-	74.8	80.7	77.7	-	72.1	77.7	75.9
	PLOME (Liu et al., 2021)	-	77.4	81.5	79.4	-	75.3	79.3	77.2
	MLM-phonetics (Zhang et al., 2021)	-	77.5	83.1	80.2	-	74.9	80.2	77.5
	ReaLiSe (Xu et al., 2021)	84.7	77.3	81.3	79.3	84.0	75.9	79.9	77.8
	UGMSC(ReaLiSe)	85.6	78.7	82.4	80.5	84.9	77.2	81.0	79.1
	MDCSpell (Zhu et al., 2022)	-	80.8	80.6	80.7	-	78.4	78.2	78.3
	SCOPE (Li et al., 2022)	-	81.1	**84.3**	82.7	-	79.2	**82.3**	80.7
	UGMSC(SCOPE)	**87.8**	**83.9**	82.8	**83.3**	**87.1**	**82.4**	81.3	**81.9**

Table 2: Detection and correction sentence-level performance (%) on the test sets of SIGHAN, where precision (Pre.), recall (Rec.), and F1 are reported. Baseline results are directly taken from their respective literature. Following (Xu et al., 2021). Results marked by "†" are calculated by applying simple post-processing on the SIGHAN13 test set which removes all detected and corrected "的", "地", and "得" from the model output before evaluation.

4.2. Baseline Methods

We compare the UGMSC(SCOPE) and UGMSC(ReaLiSe) with the following methods: **FASpell** (Hong et al., 2019) consists of a masked language model (MLM) as a denoising autoencoder (DAE) and a decoder. **SpellGCN** (Cheng et al., 2020) learns Chinese character similarity knowledge from a fixed confusion set through graph convolutional networks (GCNs). **ReaLiSe** (Xu et al., 2021) leverages semantic, phonetic, and graphic modalities and fuses them to improve CSC task performance. **MLM-phonetics** (Zhang et al., 2021) consists of a detection module and a correction module, and then pre-trains them with phonetic features. **MDCSpell** (Zhu et al., 2022) enhances the performance of the correction network supported by a detection network. **PLOME** (Liu et al., 2021) captures phonetic and graphic features through GRU and performs pinyin prediction as an auxiliary task. **SCOPE** (Li et al., 2022) introduces a fine-grained auxiliary character pronunciation prediction task to improve the CSC task.

4.3. Main Results

We train our UGMSC(SCOPE) model with the same training settings as SCOPE, and the same procedure goes for UGMSC(ReaLiSe). The evaluation results on the SIGHAN test sets are shown in Table 2. We can see that our UGMSC(SCOPE) performs significantly better than all baseline models, our UGMSC(ReaLiSe) also performs much better than its base model ReaLiSe, just by applying our framework to the previous multimodal model with the same training settings, demonstrating the robustness of our framework. We achieve performance improvements without paying much attention to tuning the training hyper-parameters. Following SCOPE, the metrics of SCOPE and UGMSC(SCOPE) are reported after applying a simple and effective post-processing trick: constrained iterative correction during inference. In particular, SCOPE corrects a sentence by two inferences, only the corrections that appear in the position next to the position that is corrected in the previous inference, and then restores the position that is modified in each iteration.

The results of Table 2 show that the UGMSC(SCOPE) performs better than SCOPE

Model	Pre.	Rec.	F1
	Detection Level		
PLOME*	75.6	79.6	77.6
UGMSC(PLOME)[1]	75.9	**80.1**	77.9
UGMSC(PLOME)[2]	76.2	79.9	78.0
UGMSC(PLOME)[3]	**77.0**	80.0	**78.5**
	Correction Level		
PLOME*	73.2	77.2	75.1
UGMSC(PLOME)[1]	73.4	77.4	75.4
UGMSC(PLOME)[2]	74.1	77.6	75.8
UGMSC(PLOME)[3]	**75.0**	**77.8**	**76.4**

Table 3: Results (%) of the reimplemented PLOME(PLOME*) and the PLOME model applying our framework with l Transformer layers of each corrector (UGMSC(PLOME)^l) on the SIGHAN15 test set (average of 5 experiments).

Model	Pre.	Rec.	F1
	Detection Level		
ReaLiSe	77.9	78.5	78.1
UGMSC(ReaLiSe)	78.5↑	79.3↑	78.8↑
SCOPE	79.5	80.3	79.9
UGMSC(SCOPE)	81.8↑	79.1	80.4↑
	Correction Level		
ReaLiSe	76.5	77.0	76.7
UGMSC(ReaLiSe)	77.5↑	78.2↑	77.8↑
SCOPE	77.7	78.7	78.4
UGMSC(SCOPE)	80.8↑	78.1	79.4↑

Table 4: Results of UGMSC(ReaLiSe) and UGMSC(SCOPE) on all SIGHAN test sets (average of all SIGHAN test sets). Baseline results are directly taken from their respective literature.

in the precision and F1 score and significantly outperforms all other baselines on all SIGHAN test datasets in almost all metrics. We can also observe that the improvement of the UGMSC(SCOPE) model based on the SCOPE model in the correction level F1 score (+1.1/+0.8/+1.2 on SIGHAN13/14/15) is larger than the improvement in the detection level (+0.8/+0.2/+0.6 on SIGHAN13/14/15). This indicates that leveraging modalities under the guidance of uncertainty is beneficial for multimodal CSC models to correct character.

4.4. Effects of corrector layers

Our UGMSC framework introduces a hyperparameter: the layers Transformer of a corrector, i.e. l in Eq. 1. We reimplement the PLOME (Liu et al., 2021) model with PyTorch based on a public repository [4] and then apply UGMSC to the reimplemented PLOME model with different Transformer layers of each corrector. We initialize these models with pre-trained parameters provided by the official repository of the PLOME model and train these models with the same training settings. We then evaluate these models on the SIGHAN15 test set and repeat the experiment 5 times. The average metrics are shown in Table 3.

Model	Acc.	Pre.	Rec.	F1
	Detection Level			
UGMSC(SCOPE)	84.0	81.8	79.1	80.4
-wo-DCT	81.9	78.0	78.8	78.3
-wo-UMF	82.6	80.4	76.9	78.6
-wo-UC	81.0	79.6	75.1	77.2
UGMSC(ReaLiSe)	82.4	78.5	79.3	78.8
-wo-DCT	81.9	78.0	78.1	78.0
-wo-UMF	81.9	77.5	78.4	77.9
-wo-UC	81.6	76.9	78.8	77.7
	Correction Level			
UGMSC(SCOPE)	83.3	80.8	78.1	79.4
-wo-DCT	81.0	76.5	77.2	76.8
-wo-UMF	82.1	78.3	76.9	77.6
-wo-UC	80.3	78.4	74.0	76.1
UGMSC(ReaLiSe)	81.8	77.5	78.2	77.8
-wo-DCT	81.2	76.9	77.1	76.9
-wo-UMF	81.1	76.2	77.0	76.5
-wo-UC	80.8	75.6	77.4	76.4

Table 5: Results of ablation study on all SIGHAN test sets (average of all SIGHAN test sets), -wo-DCT: without direct correctors training, -wo-UMF: without uncertainty-aware modality fusion, -wo-UC: without uncertainty-aware correction.

The metrics of the reimplemented PLOME are lower than those reported in their literature (the correction level F1 score drops from 77.2% to 75.1%), which may be due to the lack of output pronunciation mapping in the pre-trained parameters they provide, or the difference between TensorFlow and PyTorch.

Although UGMSC(PLOME) is statistically better, the improvements are somehow limited with fewer layers of the correctors, we attribute this to the fact that the multimodal encoders of the PLOME model are unaware of the positional information, and therefore the correctors need to learn the sentence-level features themselves. The scores of the model applying our framework increase with the number of layers of modality correctors, proving the validity of the modality correctors and the

[4] https://github.com/Zhouyuhao97/PLOME_finetune_pytorch

I	老	师	就	进	教	师
R	老	师	就	请	教	室
O	老	师	就	进	教	室
u^s	0.00	0.00	0.00	0.97	0.00	0.00
u^p	0.00	0.00	0.00	0.96	0.00	0.00
u^g	0.00	0.00	0.00	0.95	0.00	0.00
u^j	0.00	0.00	0.00	0.98	0.00	0.00
ω^s	0.62	0.02	0.25	0.04	0.01	0.05
ω^p	0.08	0.01	0.12	0.30	0.15	0.57
ω^g	0.11	0.93	0.56	0.38	0.78	0.00
ω^j	0.19	0.04	0.07	0.29	0.05	0.38

Table 6: Visualization of the key values during the inference of ReaLiSe and UGMSC(ReaLiSe) of case 1. "I" is the input sentence. "R" is the output of ReaLiSe. "O" is the output of UGMSC(ReaLiSe) (also the ground truth). s, p, g, and j represent the semantic, phonetic, graphic, and joint corrector, respectively. u is the normalized detection level entropy in Eq.2. ω are the prediction weights in Eq.6. misspelled and correct characters are marked in red/blue.

superiority of our framework.

4.5. Model-Agnostic Experiment

As shown in Table 3 and Table 4, we successfully employ the UGMSC framework in 3 advanced multimodal CSC models and outperform all the base models in almost all metrics. Compared to the ReaLiSe model, UGMSC(ReaLiSe) achieves an absolute improvement of +0.9%/+1.4% detection/correction F1 on all SIGHAN test sets with the same training settings, while UGMSC(SCOPE) gains +0.6%/+1.3% compared to the base model SCOPE, proving that our uncertainty guided multimodal framework is broadly applicable.

4.6. Ablation Study

We conduct ablation study on UGMSC(ReaLiSe) and UGMSC(SCOPE) model with the following settings: (wo-DCT) train the uncertainty-aware correction predict probability distribution $\mathbf{p}_i^u$ in Eq.6 only; (-wo-UMF) replace the uncertainty-aware modality fusion mechanism of the joint corrector with simply summation; (-wo-UC) replace the uncertainty-aware correction module with a simple summation of the prediction logits provided by all correctors. The average evaluation results of all SIGHAN test sets are shown in Table 5. We can see a significant drop in performance without these components, confirming the effectiveness of these components in our method.

I	明	天	早	上	八	点	中
R	明	天	早	上	八	点	中
O	明	天	早	上	八	点	钟
g^a	0.19	0.13	0.16	0.13	0.28	0.19	0.23
m^p	0.27	0.33	0.36	0.26	0.33	0.20	0.56
u^s	0.00	0.00	0.00	0.00	0.00	0.00	0.11
u^p	0.00	0.00	0.00	0.00	0.00	0.00	0.07
u^g	0.00	0.00	0.00	0.00	0.00	0.00	0.16
u^j	0.00	0.00	0.00	0.00	0.00	0.00	0.08
ω^p	0.02	0.65	0.25	0.06	0.04	0.01	0.69
ω^j	0.14	0.25	0.41	0.11	0.38	0.00	0.18

Table 7: Visualization of the key values during the inference of ReaLiSe and UGMSC(ReaLiSe) of case 2. m^p, g^a are the weights of phonetic modality (i.e. the values indicating the importance of modalities) given by UGMSC(ReaLiSe) and ReaLiSe respectively, m^p is mentioned in Eq.4. g^a is taken from the public code repository of the ReaLiSe model. Other notations are the same as Table 6.

I	旁	边	的	人	头	手	册
R	旁	边	的	人	投	手	册
O	旁	边	的	人	偷	手	册
g^t	1.00	1.00	1.00	1.00	1.00	1.00	1.00
m^s	0.93	0.93	0.94	0.96	0.76	0.96	0.95
g^a	0.19	0.26	0.15	0.21	0.41	0.22	0.26
m^p	0.42	0.41	0.32	0.18	0.53	0.29	0.36
g^v	0.12	0.05	0.06	0.10	0.41	0.04	0.04
m^g	0.21	0.22	0.44	0.46	0.47	0.21	0.23
u^j	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ω^j	0.53	0.02	0.12	0.01	0.95	0.09	0.06

Table 8: Visualization of the key values during the inference of ReaLiSe and UGMSC(ReaLiSe) of case 3. m^s, g^t are the weights of the semantic modality, m^g, g^v are the weights of the graphic modality, m^s, m^g, m^p is mentioned in Eq.4. g^t, g^a, g^v is taken from the public code repository of the ReaLiSe model. Other notations are the same as Table 6. and Table 7.

4.7. Case Study

Case 1 is shown in Table 6, which has a misspelled character, the 6th character "师" is misspelled, and the ReaLiSe model[5] corrects the misspelled character but gives a wrong prediction on the 4th character "进", which is not misspelled. Case 2 shows a case where ReaLiSe leaves the mistaken character "中" while our model makes a correct prediction.

Both case 1 and case 2 show the shortcomings of fusing all modalities (semantic, phonetic, graphic) to make predictions, which is inappropriate in these situations. The ReaLiSe model, which uses the selective modality fusion mechanism to fuse modalities, makes incorrect predictions on the

[5]ReaLiSe official code repository: https://github.com/DaDaMrX/ReaLiSe

character “进”, and the uncertainty metrics given by the joint corrector of our UGMSC(ReaLiSe) model for the characters “进” and in case 1 are the highest among all modality correctors, but our model makes correct predictions with the help of the uncertainty-aware correction module. Moreover, the uncertainty metrics given by the graphic corrector for the character “进” in case 1 and the uncertainty metrics given by the phonetic corrector for the character “中” in case 2 are the lowest among the other uncertainty values, indicating that there is confusing information present in other modalities, suggesting us to use multimodal information appropriately.

The case in Table 8 shows that the uncertainty-aware modality fusion mechanism performs better than the selective modality fusion mechanism of ReaLiSe in this case, because the uncertainty given by the joint corrector is very low, while the prediction results come mainly from the joint corrector (weight value is 0.95). Our model makes the correct prediction while ReaLiSe fails, and our model gives a lower gate value of the semantic modality of the wrong character “头” among others, while ReaLiSe remains very high on the contrary.

5. Conclusion

This paper proposes UGMSC, a framework that aims to appropriately leverage multimodal information under uncertainty guidance to improve the performance of the CSC task. UGMSC first estimates the uncertainty of the modalities and then incorporates these uncertainties into multimodal feature learning and correction decisions. Our framework implementation achieves great progress on the SIGHAN benchmarks compared to the base models, the results of the experiments verify the positive effects of our framework.

Acknowledgement

We would like to thank the anonymous reviewers for their insightful and valuable suggestions. This work is supported by College of Software Nankai University, Haihe Lab of ITAI and the youth program of National Science Fund of Tianjin, China (Grant No. 22JCQNJC01340).

6. Bibliographical References

Tao-Hsing Chang, Hsueh-Chih Chen, and Cheng-Han Yang. 2015. Introduction to a proofreading tool for Chinese spelling check task of SIGHAN-8. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, pages 50–55, Beijing, China. Association for Computational Linguistics.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. A study of language modeling for Chinese spelling check. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, pages 79–83, Nagoya, Japan. Asian Federation of Natural Language Processing.

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. arXiv preprint arXiv:2004.14166.

Wei-Cheng Chu and Chuan-Jie Lin. 2015. NTOU Chinese spelling check system in sighan-8 bake-off. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, pages 137–143, Beijing, China. 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.

Zijia Han, Chengguo Lv, Qiansheng Wang, and Guohong Fu. 2019. Chinese spelling check based on sequence labeling. In 2019 International Conference on Asian Language Processing (IALP), pages 373–378.

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.

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

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

Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022. Improving Chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4275–4286, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and phonologically similar characters in incorrect simplified Chinese words. In Coling 2010: Posters, pages 739–747, Beijing, China. Coling 2010 Organizing Committee.

Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. PLOME: Pre-training with misspelled knowledge for Chinese spelling correction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2991–3000, Online. Association for Computational Linguistics.

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

C. E. Shannon. 1948. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

Baoxin Wang, Wanxiang Che, Dayong Wu, Shijin Wang, Guoping Hu, and Ting Liu. 2021. Dynamic connected networks for Chinese spelling check. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2437–2446, Online. Association for Computational Linguistics.

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.

Yih-Ru Wang and Yuan-Fu Liao. 2015. Word vector/conditional random field-based Chinese spelling error detection for SIGHAN-2015 evaluation. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, pages 46–49, Beijing, China. Association for Computational Linguistics.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, listen, and see: Leveraging multimodal information helps chinese spell checking.

Shoujian Yang and Lian Yu. 2022. Cospa: An improved masked language model with copy mechanism for chinese spelling correction. In Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, volume 180 of Proceedings of Machine Learning Research, pages 2225–2234. PMLR.

Junjie Yu and Zhenghua Li. 2014. 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, pages 220–223, Wuhan, China. Association for Computational Linguistics.

Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan Wang, Zhongjun He, Yu Sun, Hua Wu, and Haifeng Wang. 2021. Correcting Chinese spelling errors with phonetic pretraining. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2250–2261, Online. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. arXiv preprint arXiv:2005.07421.

Shuiyuan Zhang, Jinhua Xiong, Jianpeng Hou, Qiao Zhang, and Xueqi Cheng. 2015. HANSpeller++: A unified framework for Chinese spelling correction. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, pages 38–45, Beijing, China. Association for Computational Linguistics.

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

7. Language Resource References

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. 2023. General and domain-adaptive chinese spelling check with error-consistent pretraining. ACM Transactions on Asian and Low-Resource Language Information Processing, 22(5):1–18.

Tseng, Yuen-Hsien and Lee, Lung-Hao and Chang, Li-Ping and Chen, Hsin-Hsi. 2015. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. Association for Computational Linguistics.

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

Wu, Shih-Hung and Liu, Chao-Lin and Lee, Lung-Hao. 2013. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. Asian Federation of Natural Language Processing.

Yu, Liang-Chih and Lee, Lung-Hao and Tseng, Yuen-Hsien and Chen, Hsin-Hsi. 2014. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. Association for Computational Linguistics.

Dataset	Method	Detection Level				Correction Level			
		Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
LEMON v2	ReaLiSe (Xu et al., 2021)	7.2	2.1	3.9	2.7	6.7	1.6	3.0	2.0
	UGMSC(ReaLiSe)	**33.0**	**15.0**	18.0	**16.3**	**31.1**	**11.8**	14.1	**12.8**
	SCOPE (Li et al., 2022)	29.4	13.1	**18.3**	15.3	27.9	10.9	**15.2**	12.7
	UGMSC(SCOPE)	13.1	3.3	5.7	4.2	12.7	2.9	4.9	3.6
ECSpell	ReaLiSe (Xu et al., 2021)	17.1	8.4	14.0	10.5	15.3	6.3	10.6	7.9
	UGMSC(ReaLiSe)	48.1	29.4	30.2	29.8	43.4	20.6	21.2	20.9
	SCOPE (Li et al., 2022)	**49.7**	**30.4**	**36.4**	**33.2**	**46.6**	**25.4**	**30.4**	**27.7**
	UGMSC(SCOPE)	26.6	11.8	16.9	13.9	25.6	10.5	14.9	12.3

Table 9: Detection and correction sentence-level performance (%) on the test sets of LEMON v2 and ECSpell, where precision (Pre.), recall (Rec.), and F1 are reported.

A. Appendix

A.1. Experiments on Other Datasets

Dataset	#Sent	Avg. Length	#Errors
SIGHAN13	1000	74.3	1224
SIGHAN14	1062	50.0	771
SIGHAN15	1100	30.6	703
LEMON v2	21734	35.2	12049
ECSpell	8172	41.8	6667

Table 10: Statistics of the SIGHAN, LEMON v2, and ECSpell test datasets, including the number of sentences, the average sentence length in tokens, and the number of errors in characters.

Datasets such as LEMON (Wu et al., 2023) and ECSpell (Lv et al., 2023) have been published recently, we directly test the checkpoints of UGMSC(SCOPE), SCOPE, UGMSC(ReaLiSe), ReaLiSe[6] on LEMON(v2) and ECSpell dataset[7] without any fine-tuning, the results are shown in Table 9. Since there are some exceptions (such as unrecognized characters, different lengths of input and target sequences, etc.) thrown during data processing, we ignore these cases when testing the models. The statistical information of the processed LEMON and ECSpell datasets is shown in Table 10.

Note that SCOPE performs much better than UGMSC(SCOPE), while UGMSC(ReaLiSe) performs much better than ReaLiSe, even than SCOPE on the LEMON v2 test dataset. We attribute this to the fact that the partial modality correctors are randomly initialized, and UGMSC(SCOPE) has no multimodal encoders, its partial modality correctors have to learn directly from the embeddings, therefore the partial modality correctors are not well trained compared to the SCOPE model, which is a fully pre-trained model, it's reasonable that UGMSC(SCOPE) performs poorly on new domain data.

UGMSC(ReaLiSe) learns from the hidden states provided by pre-trained multimodal encoders, so the partial modality correctors can be treated as partially pre-trained, so UGMSC(ReaLiSe) outperforms ReaLiSe, as expected.

[6]The checkpoint of ReaLiSe is downloaded from the official repository

[7]We use the data directly from https://github.com/gingasan/lemon