electronics

MDPI

Article

Online Mongolian Handwriting Recognition Based on Encoder–Decoder Structure with Language Model

Daoerji Fan *, Yuxin Sun, Zhixin Wang and Yanjun Peng

College of Electronic Information Engineering, Inner Mongolia University, Hohhot 010021, China; 32156020@mail.imu.edu.cn (Y.S.); 32156005@mail.imu.edu.cn (Z.W.); 32256046@mail.imu.edu.cn (Y.P.)
* Correspondence: fandaoerji@imu.edu.cn; Tel.: +86-133-1489-4340

Abstract: Mongolian online handwriting recognition is a complex task due to the script's intricate characters and extensive vocabulary. This study proposes a novel approach by integrating a pre-trained language model into the sequence-to-sequence(Seq2Seq) + attention mechanisms(AM) model to enhance recognition accuracy. Three fusion models, including former, latter, and complete fusion, are introduced, showing substantial improvements over the baseline model. The complete fusion model, combined with synchronized language model parameters, achieved the best results, significantly reducing character and word error rates. This research presents a promising solution for accurate Mongolian online handwriting recognition, offering practical applications in preserving and utilizing the Mongolian script.

Keywords: Mongolian script; online handwriting recognition; pre-trained language model; fusion model

Citation: Fan, D.; Sun, Y.; Wang, Z.; Peng, Y. Online Mongolian Handwriting Recognition Based on Encoder–Decoder Structure with Language Model. *Electronics* **2023**, *12*, 4194. https://doi.org/10.3390/electronics12204194

Academic Editors: Morgado Dias, Fabio Mendonca and Sheikh Shanawaz Mostafa

Received: 19 September 2023
Revised: 7 October 2023
Accepted: 8 October 2023
Published: 10 October 2023

1. Introduction

The traditional Mongolian script, known as "Mongol Bichig", is a unique writing system with a vertical layout, primarily used for the Mongolian language. It features complex, cursive characters with ligatures and pictorial elements. Historically, it has preserved cultural heritage and remains important in contemporary Mongolia for newspapers, official documents, and education.

Mongolian employs a phonetic script similar to English, but its writing style is distinct. In Mongolian, words are written vertically from top to bottom, with all the letters fused together to create a vertical backbone, as shown in Figure 1. Letters are categorized as initial, medial, or final based on their position within a word. The same letter, when placed differently, can undergo shape transformations, which can significantly impact the accuracy of the Mongolian handwriting recognition model.

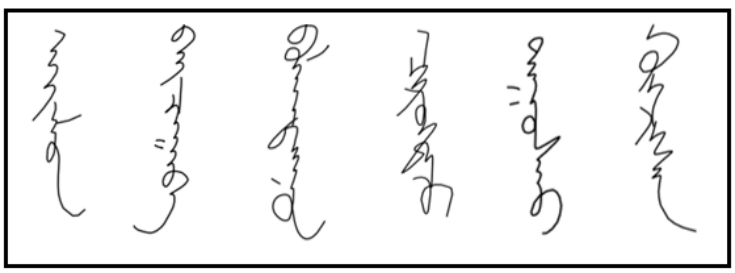

Figure 1. Mongolian handwritten samples.

Mongolian is classified as an agglutinative language, characterized by its extensive vocabulary. The primary feature of agglutinative languages is their ability to derive numerous other words from a single root word. Based on incomplete statistics, commonly used

Mongolian words (considering only root words) amount to approximately 60,000. When considering all possible derivations, Mongolian may encompass millions of words [1].

Clearly, in the training of Mongolian handwriting recognition systems, the vocabulary within the training dataset represents only a small fraction of the entire lexicon, making it highly likely to encounter out-of-vocabulary (OOV) words. This results in the model having a relatively high recognition accuracy for in-vocabulary words but a significantly lower recognition capability for OOV words. To address this issue, the common approach is to incorporate a post-processing module based on a dictionary or language model after the recognition model [2,3]. However, in this scheme, the recognition model and the post-processing model are independent of each other, cannot be jointly optimized, and do not support end-to-end recognition.

In response to the above problems, this article proposes a novel fusion model for online Mongolian handwriting recognition. In summary, we trained a character-level Mongolian language model using large-scale Mongolian word corpora and incorporated the pre-trained language model into a traditional Encoder–Decoder handwriting recognition model. The decoder not only receives the writing features from the encoder but also receives guidance information from the language model, using the fused information as a basis for decoding and judgment, in order to improve the accuracy of the recognition of OOV. During Encoder–Decoder model training, the parameters of the pre-trained language model can also be fine-tuned. This not only achieves joint tuning of the recognition model and the language model but also achieves end-to-end recognition of Mongolian.

2. Related Work

In the realm of Mongolian language processing and recognition, several significant contributions and advancements have emerged in recent years. Pan et al. [4] introduced the MOLHW dataset, a substantial Mongolian online handwriting dataset containing 164,631 samples encompassing 40,605 Mongolian words, curated by 200 native Mongolian-speaking college students. This dataset has served as a foundational resource for research in the field.

Cui et al. [2] presented an end-to-end neural network model tailored to irregularly printed Mongolian text recognition. Their comprehensive approach spans from image input to text output, enabling direct detection and extraction of Mongolian text from images. This end-to-end methodology has significantly enhanced the accuracy and efficiency of recognizing irregularly printed Mongolian text.

Sun et al. [5] introduced the Mongolian Generator (MG), a novel approach employing Generative Adversarial Networks (GANs) for automated Mongolian handwritten word image generation. MG excels in producing highly detailed and accurate word images, accommodating diverse writing styles and text content. The integration of perceptual adversarial loss further enhances the realism of generated images. This innovation simplifies Mongolian handwritten image synthesis, benefiting both researchers and practitioners.

Fan et al. [3] addressed the challenges in Mongolian language processing arising from its agglutinative nature. They proposed a sub-word-based language model to mitigate high out-of-vocabulary rates and data sparsity. The system encompasses three key components: handwritten image preprocessing, image-to-grapheme mapping, and LM decoding, collectively contributing to improved Mongolian text processing.

In [6], a segmentation-free approach for efficient Optical Character Recognition (OCR) of Mongolian text was presented. The end-to-end model directly extracts features from input word images, surpassing glyph segmentation-based methods in performance. It also effectively handles out-of-vocabulary words, further enhancing Mongolian text recognition.

Da et al. [7] introduced an end-to-end model for Traditional Mongolian online handwritten word recognition. This model combines a bidirectional Long Short-Term Memory (LSTM) network with a Connectionist Temporal Classification (CTC) network. The core of the model is the bidirectional LSTM network, augmented by the CTC network, facilitating efficient label recognition for input sequences. Additionally, the study delves into error analysis, addressing a less-explored area in the context of online handwritten Mongolian recognition.

Fan Yang et al. [8] proposed a segmentation-free, lightweight network structure for online handwritten Mongolian character recognition. Their model utilizes a one-dimensional CNN for feature extraction, followed by a Bidirectional Long Short-Term Memory(BiLSTM) Seq2Seq model for the conversion of variable-length sequences.

Wei et al. [9] presented an end-to-end model for offline Mongolian word recognition. This model employs a sequence-to-sequence architecture with attention mechanisms, featuring two LSTMs and an attention network. Experimental results highlight the model's superior performance compared to state-of-the-art methods.

In [10], an additional Mongolian online handwriting dataset, MRG-OHMW, containing 946 Mongolian words was introduced, co-created by 300 Mongolian volunteers. Ji Liu et al. [11] used the MRG-OHMW dataset for data augmentation by placing handwritten Mongolian words under a grid at different locations on a canvas. They employed two feature combination methods and a multi-classification combination strategy to perform recognition using CNNs, enriching the information of Mongolian shapes.

Fan et al. [12] propose a hybrid model combining hidden Markov models (HMMs) and deep neural networks (DNNs) for Mongolian offline handwriting recognition. The concept of Mongolian grapheme code decomposition is first introduced in this work and is used as the smallest modeling unit. In our study, grapheme codes are also employed as Mongolian text encoding, with the definition of glyph codes provided in Figure 2.

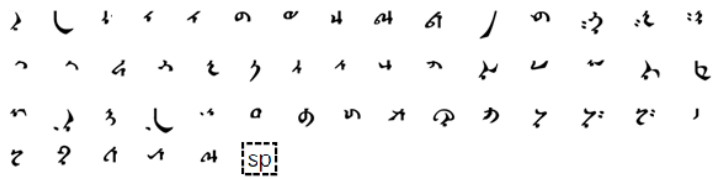

Figure 2. Grapheme code.

These studies collectively represent significant advancements and contributions in the domain of Mongolian language processing and recognition, addressing various challenges and pushing the boundaries of research in this field.

Encoder–decoder models have gained widespread adoption in the field of online handwritten text recognition, as evidenced by recent studies [13–15]. These models, equipped with attention mechanisms, have proven effective in converting handwritten trajectories into textual outputs. The attention mechanism is a fundamental component in the field of natural language processing and deep learning. It has gained significant popularity due to its ability to enhance the performance of various sequence-to-sequence tasks, including machine translation, text summarization, and speech recognition [16].

The integration of pre-trained language models and encoder–decoder architectures is not a novel concept, as demonstrated in prior research. In [17], the effective incorporation of pre-trained masked language models such as BERT into encoder–decoder models for grammatical error correction (GEC) was explored. Similarly, ref. [18] enhanced Attention-based Encoder–Decoder (AED) models by integrating external language models (LMs) and adopted a Bayesian approach, incorporating the internal language model. Additionally, ref. [19] addressed the challenge of integrating external language models (LMs) into end-to-end automatic speech recognition (ASR) systems, particularly when no clear divisions exist between acoustic and language models.

Building upon this existing research, our work introduces a novel application: the utilization of pre-trained language models for Mongolian script cursive handwriting recognition tasks, marking the first of its kind in this context.

3. Method

Encoder–decoder models in online handwritten text recognition work by taking handwritten trajectory as input, encoding them into a latent representation using the

encoder, and then decoding this representation to produce readable textual output. This procedure involves recognizing and comprehending the handwritten characters and their spatial arrangement, enabling tasks such as text conversion, translation, and handwriting generation, making them valuable tools in the field of handwritten text processing.

This paper enhances the traditional Gate Recurrent Unit(GRU) based encoder–decoder structure by incorporating a language model, achieving an end-to-end Mongolian handwritten text recognition model, as illustrated in Figure 3. In traditional encoder–decoder models, the decoder initially predicts the next character after the start symbol <SOS> based on the encoder's output. Subsequently, this predicted character becomes the input for predicting the subsequent characters, continuing until the end symbol <EOS> is predicted.

To enhance prediction accuracy, an attention module is often integrated into the encoder–decoder architecture. The core idea behind the attention mechanism involves calculating a set of attention weights that determine how much attention should be assigned to each input element. The attention weights are computed using a scoring function, often based on the similarity between the current decoder state and the encoder outputs. The scoring function utilized in this paper is as shown in Equation (1):

$$\text{Attention Weight}_i = \frac{\exp\left(\mathbf{h}_x^T \cdot \mathbf{h}_y^i\right)}{\sum_j \exp\left(\mathbf{h}_x^T \cdot \mathbf{h}_y^j\right)} \tag{1}$$

where $\mathbf{h}_x$ represents the encoder output and $\mathbf{h}_y$ represents the current hidden state of decoder.

This module ensures that, when predicting the current character, the model focuses on relevant portions of the overall writing trajectory, thereby improving its overall performance. The decoder itself operates in a manner very similar to a language model, where it predicts the next most likely character based on the current character. However, when understood as a language model, the decoder is trained using only the vocabulary present in the training dataset, which is not as comprehensive as training a language model.

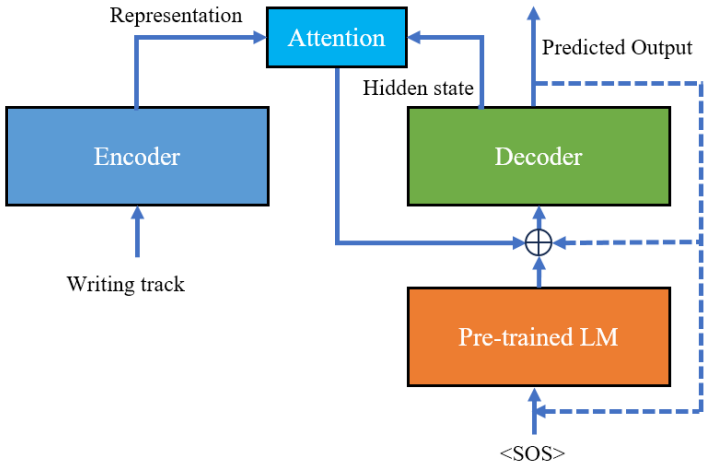

Figure 3. The overview framework of the proposed model.

We all know that when we see most of a word, it is often easy to guess the remaining part. For example, when the current prediction is "*schoo*", there's a high likelihood that the next character is "*l*". In handwriting recognition, this kind of prediction can be made without relying on the handwriting information. However, in traditional encoder–decoder models for handwriting recognition, the aforementioned contextual semantic information is not fully leveraged, especially in languages with extensive vocabularies such as Mongolian. To achieve this, a model combining a pre-trained character-level language model and a decoder has been proposed. This model is used to integrate contextual semantic information

with handwriting trajectory data to predict the next character. Subsequently, we will provide a comprehensive overview of each module.

3.1. Character-Level Language Model

The language model, an essential and pivotal component in natural language processing, can be trained using extensive, unsupervised text data to independently acquire knowledge about word associations. Its primary role is to calculate the likelihood of a sentence composed of multiple words, enabling it to assess the sentence's fluency. A well-trained language model can furnish the likelihood of a grammatical word within a sentence, which is valuable for generating text and predicting the next word. Because Mongolian is an agglutinative language with an extensive vocabulary, utilizing a character-level language model is the optimal choice.

The Gated Recurrent Unit (GRU) model is commonly applied in language models to capture sequential dependencies in text data. It is a variation of recurrent neural networks (RNNs) designed to address the vanishing gradient problem and improve the training of deep networks. GRU models excel in tasks such as natural language processing (NLP) and language modeling due to their ability to capture long-range dependencies while mitigating some of the challenges faced by traditional RNNs, such as the exploding and vanishing gradient problems. Mikolov proposed the use of RNNs in language models in 2010 [20], and subsequent researchers have obtained good results using such models [21]. We propose a GRU-based language model that can be divided into three parts, as shown in Figure 4.

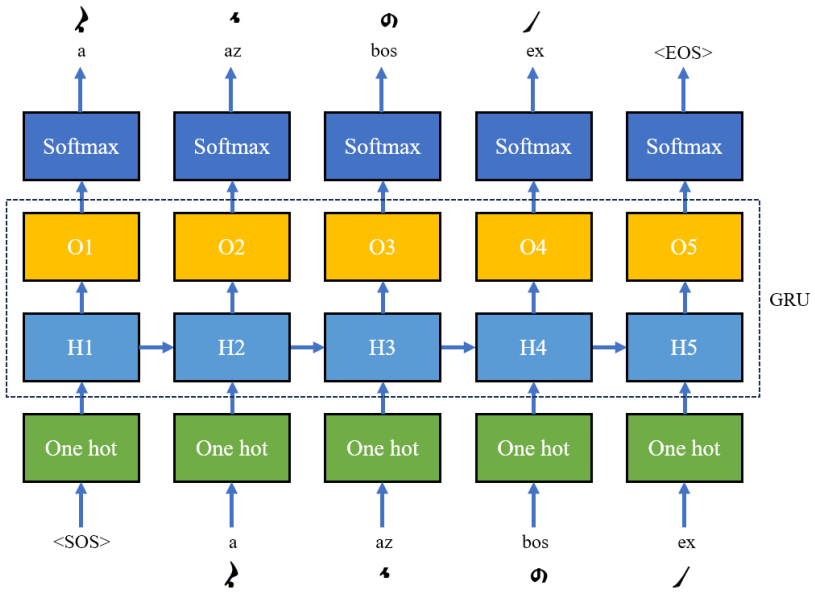

Figure 4. Structure of language model.

The first part is an encoding layer that converts grapheme codes into one-hot vectors of fixed dimensionality to facilitate feature learning and computation by the neural network. The second part is a unidirectional GRU network that learns features for each input one-hot vector, and subsequently obtains a prediction result. The language model cannot use a bidirectional GRU because text generation can only be generated from front to back, which is unidirectional. The third part is a softmax classification module to classify the predicted output of the GRU.

In a character-level language model, the smallest unit is not a word but rather a character. For example, assume a Mongolian word corresponds to a grapheme code of "a az bos ex". We need to add the start of sentence <SOS> before the first code, and use "<SOS> a az bos ex" as the input sequence of the model. We add the end of sentence <EOS> after the last code, and use "a az bos ex <EOS>" as the output sequence.

The input sequence is one-hot processed and fed into the GRU network, and the output distribution is calculated for each step t to predict the probability distribution of each character. The loss function on the tth step is the cross-entropy between the predicted probability distribution $\hat{y}^{(t)}$ and the next real word $y^{(t)}$, as shown in Equation (2):

$$J^{(t)}(\theta) = CE\left(y^{(t)}, \hat{y}\right) = -\sum_{w \in V} y_w^{(t)} \log \hat{y}_w^{(t)} = -\log \hat{y}_{x_{t+1}}^{(t)}. \tag{2}$$

where V is corpus, $w \in V$ is a word, and $x \in W$ is a character. The results of Equation (2) are averaged to obtain the overall loss of the training set, as shown in Equation (3):

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T} J^{(t)}(\theta) = \frac{1}{T}\sum_{t=1}^{T} -\log \hat{y}_{x_{t+1}}^{(t)}. \tag{3}$$

3.2. Baseline Model

The baseline model employed in this paper is a sequence-to-sequence encoder–decoder model based on GRU and attention mechanism (Seq2Seq + AM). The encoder is a bidirectional GRU network consisting of forward and backward GRUs, with the forward GRU reading the input sequence $x = (x_1, x_2, \cdots, x_T)$ sequentially to generate the hidden state vector $(\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_T)$. The backward GRU reads the input sequence from the reverse direction, generates a hidden state vector $(\overleftarrow{h}_1, \overleftarrow{h}_2, \cdots, \overleftarrow{h}_T)$, and connects the hidden state vectors in both directions at each time step, $h_i = \left[\vec{h}_i, \overleftarrow{h}_i\right]$. Each state vector h_i corresponds to the information of the ith data and its surrounding data of the input sequence.

The decoder is constructed with a unidirectional GRU network. At each moment t, the attention mechanism decides which hidden state vectors are most relevant. The relevance weight parameter α_t of the tth word is calculated by a forward neural network f, whose inputs are the hidden state h_i of the encoder, the hidden state s_{t-1} of the decoder at the previous moment, and the output y_{t-1} of the decoder at the previous moment, as shown in Equation (4):

$$\alpha_t = f(h_i, s_{t-1}, y_{t-1}). \tag{4}$$

α_t is used to obtain the contextual information vector c_t of the tth word by Equation (5),

$$c_t = \sum_{i=1}^{T} \alpha_t h_i. \tag{5}$$

The hidden state of the decoder at the current moment is calculated by the context information vector c_t, the hidden state s_{t-1} of the decoder at the previous moment, and the predicted output y_{t-1} at the previous moment by Equation (6). Then, a fully connected layer is used to generate the output y_t of the current moment, as shown in Equation (7), and finally, the probability distribution of the output is obtained by the softmax activation function and the maximum probability is selected as the predicted character,

$$s_t = f_r(c_t, s_{t-1}, y_{t-1}) \tag{6}$$

$$y_t = W s_t + b, \tag{7}$$

where f_r is the unidirectional GRU network structure, W is the weight matrix of the linear layer, and b is the bias vector.

3.3. Fusion Model

The primary breakthrough in this paper lies in the incorporation of a pre-trained language model into the Seq2Seq + AM model, which we term the "fusion model". In this procedure, the language model underwent independent training on a substantial corpus, and the top-performing model was chosen based on perplexity (PPL) scores. Following

this, the pre-trained language model was integrated into the baseline model and co-trained with the recognition model using a handwritten dataset. Building upon the interaction between the language model and the decoder, we have introduced three fusion approaches: the former model, the latter model, and the complete fusion model.

3.3.1. Former Fusion Model

In the former fusion model, the pre-trained language model predicts the probability of the next character based on the historical inputs. This probability distribution is then incorporated as a part of the decoder's input, as illustrated in Figure 5. The input of the decoder of the original model at the tth time consists of three parameters: the context vector c_t^{Seq}, the hidden state s_{t-1}^{Seq} of the decoder at the previous time, and the predicted output y_{t-1}^{Seq} at the previous time. In the former fusion model, our goal is for the language model to continuously supply the decoder with semantic information related to Mongolian words as it operates. The output y_t^{LM} of the language model contains the prediction information of the language model for the next character, so the context vector c_t^{Seq} is spliced with the output y_t^{LM} of the language model, and this is used as the input of the decoder, together with the other two parameters, to calculate the hidden state s_t at the tth time by Equation (8). The output y_t is then obtained by Equation (9),

$$s_t^{FM_1} = f_r\left(c_t^{Seq} + y_t^{LM}, s_{t-1}^{Seq}, y_{t-1}^{Seq}\right) \tag{8}$$

$$y_t^{FM_1} = W s_t^{FM_1} + b. \tag{9}$$

Because this model fuses the Seq2Seq + AM model with the language model before the decoder works, it is called the former fusion model.

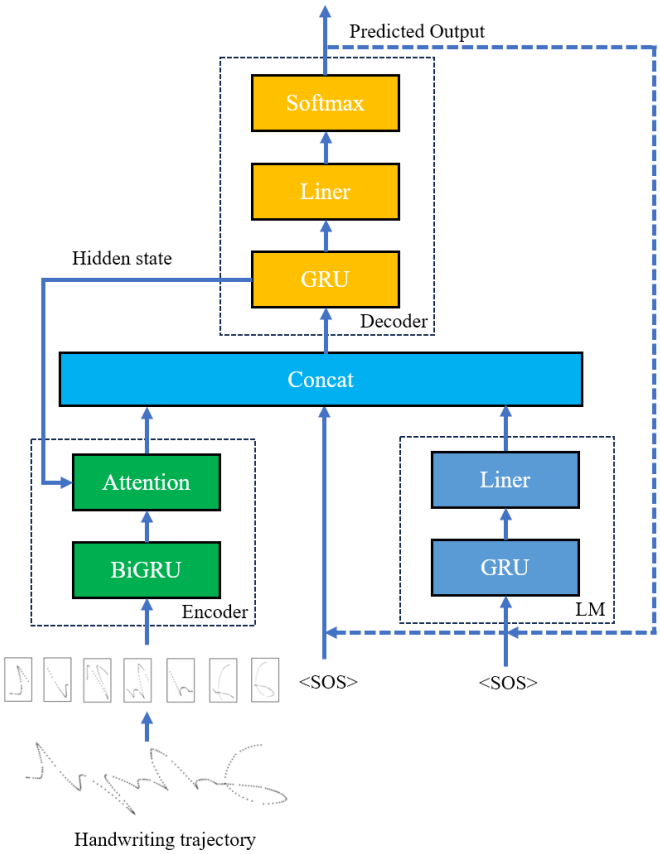

Figure 5. Structure of former fusion model.

3.3.2. Latter Fusion Model

In the latter fusion model, both the decoder and the language model receive the current character and operate independently. However, before making predictions, the outputs of the decoder and the language model are concatenated and fed into a linear layer for prediction. In the baseline model, the state s_t^{Seq} of the decoder at the tth time is fed into the fully connected layer to obtain the output y_t^{Seq} by Equation (10). We want the language model to provide semantic information to the Seq2Seq + AM model at the time of prediction, and the output of the language model has information about the word composition of the grapheme code, so we concat the output y_t^{LM} of the language model and the state s_t^{Seq}, and send the result to the fully connected layer to obtain the overall output y_t by Equation (11),

$$s_t^{Seq} = f_r\left(c_t^{Seq}, s_{t-1}^{Seq}, y_{t-1}^{Seq}\right) \tag{10}$$

$$y_t^{FM_2} = W\left(y_t^{LM} + s_t^{Seq}\right). \tag{11}$$

Because the model fuses the Seq2Seq + AM model with the language model after the decoder works, it is called the latter fusion model. Its structure is shown in Figure 6.

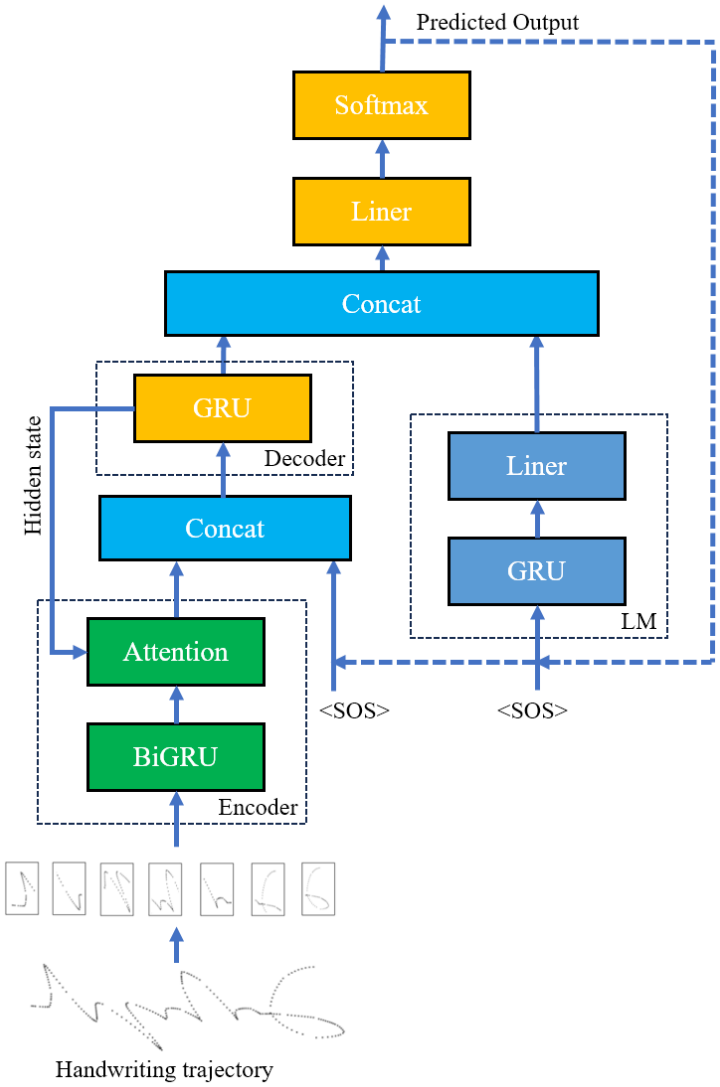

Figure 6. Structure of latter fusion model.

3.3.3. Complete Fusion Model

The above-mentioned two fusion models, respectively, let the language model provide semantic information before and after the decoding of the Seq2Seq + AM model, and both can theoretically improve its generalization ability. We combine these approaches, fusing both before and after the decoder works, as shown in Equations (12) and (13). This is called a complete fusion model. Its structure is shown in Figure 7.

$$s_t^{FM_3} = f_r\left(c_t^{Seq} + y_t^{LM}, s_{t-1}^{Seq}, y_{t-1}^{Seq}\right) \tag{12}$$

$$y_t^{FM_3} = W\left(y_t^{LM} + s_t^{FM_3}\right) \tag{13}$$

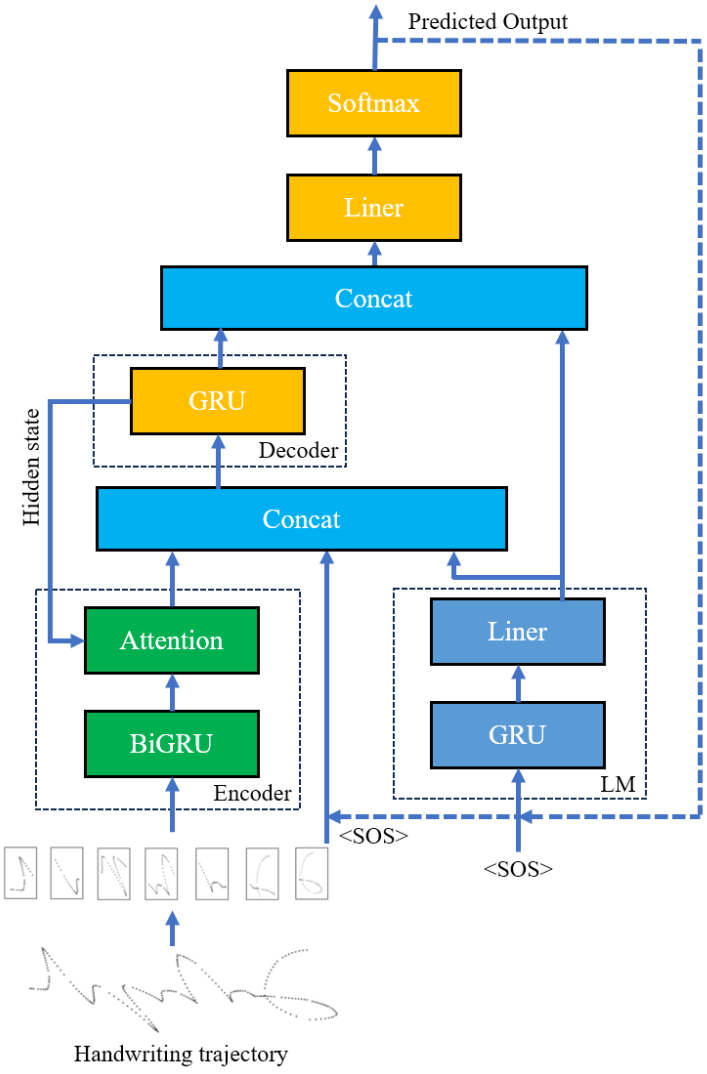

Figure 7. Structure of complete fusion model.

4. Experimental Results

4.1. Dataset

The Mongolian online handwriting dataset used in this article is called MOLHW [4]. The MOLHW dataset focuses on traditional Mongolian word-level online handwriting. This dataset comprises 164,631 handwritten Mongolian word samples contributed by 200 different writers. It encompasses over forty thousand commonly used Mongolian words, carefully selected from a substantial Mongolian corpus. The coordinate points for these words were gathered through a dedicated mobile application where volunteers wrote

out the designated words. The dataset was partitioned into training, validation, and test sets in a 7:1:2 ratio.

The corpus for training the language model in this article comes from Mongolian electronic documents, including 1,641,488 Mongolian words, which is 40 times the vocabulary size of MOLHW, and each Mongolian word is represented using a grapheme code. This dataset was likewise partitioned into training, validation, and test sets following a 7:1:2 ratio.

4.2. Evaluation Index

The fusion model was evaluated using the average char error count (ACEC) and word error rate (*WER*). ACEC is the edit distance between the predicted and target sequences. The edit distance determines the similarity of two strings, i.e., the minimum number of operations required to convert from one string to the other, including the three operations of inserting character I, deleting character D, and replacing character S. The edit distances of all samples are counted and averaged to obtain the ACEC, which is intuitively the average number of incorrect characters per word. If a deletion, insertion, or replacement occurs, the sample is considered to be incorrectly identified.

WER is calculated by Equation (14):

$$WER = \frac{I + D + S}{N}, \tag{14}$$

where N is the total number of samples, and I, D, and S are the respective numbers of insert error, delete error, and replace error samples.

Since ACEC represents the average number of character errors and *WER* is the word error rate, smaller values of ACEC and *WER* in the experimental results represent higher recognition accuracy.

Language models use *perplexity* (PPL) to measure how well they predict a sample. The idea is to evaluate a language model by assigning a high or low sentence probability value to the test set. The higher the sentence probability, the lower the perplexity, which means that the model is better trained. The *perplexity* is calculated by Equation (15):

$$perplexity(S) = p(w_1, w_2, \cdots, w_m)^{-\frac{1}{m}}, \tag{15}$$

where S is the sequence of Mongolian word grapheme codes for which the probability is to be calculated, and $w_1 \sim w_m$ are the characters in the sequence.

4.3. Experimental Results of Language Model

A GRU-based language model with a batch size of 128 was trained using the RMSprop optimization algorithm [22] with a learning rate of 0.0005. The number of GRU layers and number of neurons in the hidden layer were used as hyperparameters, perplexity was used as an evaluation index, and a simple search strategy was used to adjust the hyperparameters. The optimal model was selected based on the loss in the validation set and evaluated on the test set using the optimal path decoding algorithm.

The experimental results of the language model using the search strategy to find the optimal number of hidden layer neurons are shown in Table 1. In this experiment, the number of layers of the GRU network was fixed at one, and three experiments with different numbers of neurons were conducted. The results show that the PPL performance of both the training and test sets decreases and then increases as the number of neurons increases, which indicates that the model's overfitting and generalization ability deteriorate as the number of parameters increases. Therefore, the number of hidden layer neurons was selected as 128.

Table 1. Number of neurons of language model tuning experiment result.

Number of Neurons	train_PPL	test_PPL
64	4.181	4.380
128	**3.880**	**4.236**
256	3.905	4.296

Bold font indicates optimal results.

The experimental results of the language model using the search strategy to find the optimal number of network layers are shown in Table 2. Based on the previous experiment, the number of hidden neurons per layer of GRU was fixed at 128. In the model, the number of GRU layers was searched from one to four. The experimental results show that the PPL of the test set decreases when the number of layers increases from one to three, i.e., the performance of the model improves, but when the number of layers is changed from three to four, the PPL of the test set increases due to the mismatch between the model parameters and the dataset, the performance of the model decreases, and the predicted words become more incorrect.

Table 2. Layers of language model tuning experiment result.

Layer	train_PPL	test_PPL
1	3.880	4.236
2	3.734	4.156
3	**3.607**	**4.045**
4	3.680	4.054

Bold font indicates optimal results.

Combining the two experimental findings, the hyperparameters of the optimal language model were selected. The number of layers of the GRU network was three, and the number of neurons in the hidden layer was 128.

4.4. Experimental Results of Fusion Model

This article's baseline model, Seq2Seq + AM, is entirely based on the model proposed in reference [4]. The writing trajectory consists of sequential two-dimensional coordinates representing the writing order. We employ a sliding window that moves along the writing order and combines all coordinates within the window to form a data frame. The model's hyperparameters were determined based on the optimal configuration outlined in reference [4], which consists of three layers, a hidden layer size of 128, and a sliding window size of 20.

We built an attention-based Seq2Seq baseline model and compared it with three fusion models.

Table 3 shows the experimental results of the three fusion models, which all improve the accuracy of the Mongolian online handwritten text recognition task compared to the baseline model, with the largest improvement being that of the complete fusion model, which reduces the average number of character errors from 0.473 to 0.428 and the word error rate from 24.28% to 21.05% on the test set.

Table 3. Experimental results of three fusion models.

Model	train_ACEC	train_WER	test_ACEC	test_WER
Seq2Seq + AM [4]	0.234	13.52%	0.473	24.28%
former fusion model	0.226	12.62%	0.449	21.44%
latter fusion model	0.217	11.56%	0.435	21.22%
complete fusion model	**0.202**	**10.89%**	**0.428**	**21.05%**

Bold font indicates optimal results.

When training the fusion model, neural network parameters such as the weight parameter W and bias vector b in the baseline model are updated when the losses are computed by backpropagation, while the neural network parameters in the language model remain the same as the parameters of the optimal language model obtained from the previous experiments. To address this situation, we conducted experiments to update the parameters of the language model simultaneously with the training of the fusion model, with results as shown in Table 4.

Table 4. Comparative experiment on language model parameter training.

Language Model Parameters	train_ACEC	train_WER	test_ACEC	test_WER
no further updates	0.202	10.89%	0.428	21.05%
synchronized training	**0.186**	**9.698%**	**0.409**	**20.30%**

Bold font indicates optimal results.

The experimental results show that the recognition rate of Mongolian online handwriting can be improved by synchronizing the parameters of the language model with the complete fusion model. The ACEC on the test set is reduced to 0.409, and the WER is reduced to 20.30%. This is because the semantic information is more suitable for the MOLHW dataset after the language model parameters are adjusted.

5. Discussion

The experimental results indicate that the addition of a language model led to a 3.23% reduction in the baseline model's WER. We believe that the primary reason for the performance improvement is that the language model can prevent the generation of words that do not adhere to Mongolian grammar. Mongolian language features numerous grammatical rules for word formation, where certain characters are restricted to the word's initial position, while others can only appear at the word's end. For instance, characters such as "ᠴ ᠵ ᠽ" are prohibited from appearing at the beginning of a word, while characters such as "ᠢ ᠦ ᠣ" are only allowed in the middle of a word. Traditional handwriting recognition models focus solely on extracting effective features from text images or writing trajectories, subsequently mapping them into character sequences, without the ability to automatically learn these underlying text construction rules. We believe that incorporating a pre-trained language model can effectively address this issue.

The MOLHW dataset used in this paper was only made publicly available in 2023, which is why there is relatively limited research reported on it. In a separate study, our team achieved Mongolian online handwritten recognition on the MOLHW dataset using an LSTM-CTC model [7]. The comparison of our approach with the LSTM-CTC model is presented in Table 5. The LSTM-CTC model by Tengda et al. [7] achieved a training ACEC of 0.347 and a training WER of 21.432%. During testing, it achieved a test ACEC of 0.528 and a test WER of 30.14%. Our model outperformed the others, achieving the best results. During training, our model achieved an ACEC of 0.202 and a WER of 10.89%. During testing, it achieved an ACEC of 0.428 and a WER of 21.05%. These results demonstrate the superior performance of our approach in reducing both character error and word error rates compared to the LSTM-CTC and Seq2Seq + AM models.

Table 5. Comparison with other studies.

Model	train_ACEC	train_WER	test_ACEC	test_WER
LSTM-CTC [7]	0.347	21.432%	0.528	30.14%
Seq2Seq + AM [4]	0.234	13.52%	0.473	24.28%
Ours	**0.202**	**10.89%**	**0.428**	**21.05%**

Bold font indicates optimal results.

This study has several limitations: (1) The hyperparameter tuning for the language model was performed using a simple grid search strategy, and more sophisticated optimization methods were not explored. (2) The baseline model was based on the reference [4] without further investigation into potentially better model parameters. (3) While the model's recognition performance has improved, it has not yet reached the level required for practical end-to-end applications. In future research, we need to not only optimize model parameters but also analyze the causes of misidentification to discover methods for further improving accuracy. Specifically, we will delve into an in-depth investigation of the challenges associated with recognizing visually similar Mongolian characters.

In conclusion, as summarized above, employing a pre-trained language model to provide contextual semantic information for encoder–decoder-based handwriting recognition models proves to be an effective approach for addressing the issue of a vast vocabulary.

6. Conclusions

This research aims to improve Mongolian online handwriting recognition by incorporating a language model into the Seq2Seq + AM model. Mongolian script, known as "Mongol Bichig," presents unique challenges due to its complex characters and extensive vocabulary. This study introduces fusion models that integrate the language model with the decoder, enhancing recognition accuracy.

Experimental results on the MOLHW dataset demonstrate significant improvements. The complete fusion model, when coupled with synchronized language model parameters, reduces character and word error rates, making it an effective tool for Mongolian online handwriting recognition.

In conclusion, this study introduces an innovative approach to boost the accuracy of Mongolian online handwriting recognition. The fusion models, particularly the synchronized complete fusion model, show promise for practical applications, benefiting Mongolian script-based handwritten text recognition.

Author Contributions: Conceptualization, methodology, writing—review and editing D.F.; software, validation, Y.S.; writing—original draft preparation, Z.W.; software Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundation of China (Grant No. 61763034) and the National Natural Science Foundation of the Inner Mongolia Autonomous Region (Grant No. 2020MS06005).

Data Availability Statement: Data were obtained from kaggle and are available http://www.kaggle.com/datasets/fandaoerji/molhw-ooo (accessed on 1 July 2023) with the permission of kaggle.

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

References

1. Lu, M.; Bao, F.; Zhang, H.; Gao, G. The image and ground truth dataset of Mongolian movable-type newspapers for text recognition. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2023**, 1–13. [CrossRef]
2. Cui, S.; Su, Y.; Qing dao er ji, R.; Ji, Y. An end-to-end network for irregular printed Mongolian recognition. *Int. J. Doc. Anal. Recognit. (IJDAR)* **2022**, *25*, 41–50. [CrossRef]
3. Fan, D.; Gao, G.; Wu, H. Sub-word based mongolian offline handwriting recognition. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 246–253.
4. Pan, Y.; Fan, D.; Wu, H.; Teng, D. A new dataset for mongolian online handwritten recognition. *Sci. Rep.* **2023**, *13*, 26. [CrossRef] [PubMed]
5. Sun, S.; Wei, H. A mongolian handwritten word images generation approach based on generative adversarial networks. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–8.
6. Wang, W.; Wei, H.; Zhang, H. End-to-end model based on bidirectional lstm and ctc for segmentation-free traditional mongolian recognition. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8723–8727.
7. Teng, D.; Fan, D.; Bai, F.; Pan, Y. End-to-End Model Based on Bidirectional LSTM and CTC for Online Handwritten Mongolian Word Recognition. In Proceedings of the 2022 12th International Conference on Information Science and Technology (ICIST), Kaifeng, China, 14–16 October 2022; pp. 271–275.

8. Yang, F.; Bao, F.; Gao, G. Online handwritten Mongolian character recognition using CMA-MOHR and coordinate processing. In Proceedings of the 2020 International Conference on Asian Language Processing (IALP), Kuala Lumpur, Malaysia, 4–6 December 2020; pp. 30–33.
9. Wei, H.; Liu, C.; Zhang, H.; Bao, F.; Gao, G. End-to-end model for offline handwritten mongolian word recognition. In Proceedings of the Natural Language Processing and Chinese Computing: 8th CCF International Conference (NLPCC 2019), Dunhuang, China, 9–14 October 2019; Proceedings, Part II 8; Springer: Berlin/Heidelberg, Germany, 2019; pp. 220–230.
10. Ma, L.L.; Liu, J.; Wu, J. A new database for online handwritten Mongolian word recognition. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 1131–1136.
11. Liu, J.; Ma, L.L.; Wu, J. Online handwritten Mongolian word recognition using MWRCNN and position maps. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 60–65.
12. Daoerji, F.; Guanglai, G. DNN-HMM for large vocabulary Mongolian offline handwriting recognition. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 72–77.
13. Parres, D.; Paredes, R. Fine-Tuning Vision Encoder–Decoder Transformers for Handwriting Text Recognition on Historical Documents. In Proceedings of the International Conference on Document Analysis and Recognition, San Jose, CA, USA, 21–26 August 2023 ; Springer: Berlin/Heidelberg, Germany, 2023; pp. 253–268.
14. Kass, D.; Vats, E. AttentionHTR: Handwritten text recognition based on attention encoder-decoder networks. In Proceedings of the International Workshop on Document Analysis Systems, La Rochelle, France, 22–25 May 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 507–522.
15. Lin, Q.; Wang, C.; Bi, N.; Suen, C.Y.; Tan, J. An Encoder-Decoder Approach to Offline Handwritten Mathematical Expression Recognition with Residual Attention. In Proceedings of the International Conference on Pattern Recognition and Artificial Intelligence, Paris, France, 1–3 June 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 335–345.
16. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
17. Kaneko, M.; Mita, M.; Kiyono, S.; Suzuki, J.; Inui, K. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. *arXiv* **2020**, arXiv:2005.00987.
18. Zeineldeen, M.; Glushko, A.; Michel, W.; Zeyer, A.; Schlüter, R.; Ney, H. Investigating methods to improve language model integration for attention-based encoder-decoder asr models. *arXiv* **2021**, arXiv:2104.05544.
19. Meng, Z.; Parthasarathy, S.; Sun, E.; Gaur, Y.; Kanda, N.; Lu, L.; Chen, X.; Zhao, R.; Li, J.; Gong, Y. Internal language model estimation for domain-adaptive end-to-end speech recognition. In Proceedings of the 2021 IEEE Spoken Language Technology Workshop (SLT), Shenzhen, China, 19–22 January 2021; pp. 243–250.
20. Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association. Makuhari, Chiba, Japan, 26–30 September 2010; Volume 2, pp. 1045–1048.
21. Masumura, R.; Asami, T.; Oba, T.; Sakauchi, S.; Ito, A. Latent words recurrent neural network language models for automatic speech recognition. *IEICE Trans. Inf. Syst.* **2019**, *102*, 2557–2567.
22. Zou, F.; Shen, L.; Jie, Z.; Zhang, W.; Liu, W. A sufficient condition for convergences of adam and rmsprop. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11127–11135.