Historical Corpora Correlation based on RNN and DCNN

Wei Lin¹,a, Zhaoquan Lin²,b
¹College of Physics & Information Engineering of Fuzhou University, Fuzhou, Fujian P.R. China
²College of Physics & Information Engineering of Fuzhou University, Fuzhou, Fujian P.R. China
¹email: mqks@fzu.edu.com, bemail: 1243611238@qq.com
Corresponding author’s e-mail: mqks@fzu.edu.com

Abstract. Correcting historical corpora in digital version is a crucial task for the historical research, however, scan quality, book layout, visual character similarity can affect the quality of the recognizing. OCR is at the forefront of digitization projects for cultural heritage preservation. The main task is to identify characters from their visual form into their textual representation. In this paper, we propose a model combining recurrent neural network (RNN) and deep convolutional network (DCNN) to correct OCR transcription errors. The experiment on a historical book corpus in German language shows that the model is very robust in capturing diverse OCR transcription errors greatly.

1. Introduction
Historical Corpus’s quality is extremely crucial for historians’ research. However, due to the fact that common historical resources are not well protected, there are many ways leading to the word and character transcription errors, which greatly increased the complexity of this task.

In the past few years, there are many research done about this topic. The most common method is Post-hoc correction[3]. This method takes the OCR transcribed text as input and output its corrected version according to the error-free ground-truth transcription. So that the model’s recognizing ability can be improved by this strategy of training. For better improving the robustness of the model, its composers propose the multi-input attention to leverage redundancy among textual snippets for correction.

However, this method’s weakness is quite obvious. Manually acquiring such ground-truth is highly expensive and furthermore, typically, historical corpora does not contain redundant information. Another weakness is that there are many characters types existing in various historical corpus. By this method of training, the models’ generalization ability can not be guaranteed.

In this paper, we propose an Encoder-Decoder architecture. Encoder-Decoder architecture is combined with two ends, which are encoder and decoder[6]. Encoder converts the input sequence into vector with fixed length. The vector which is generated by encoder can express the overall meaning of all the input sequence. Correspondingly, decoder takes the responsibility of decoding this vector back into the same format of the input sequence. With comparing the difference of these two sequences which are generated successively, we can iterate to improve the model’s ability.

Usually, encoder-decoder structure can use Recurrent Neural Networks/Bidirectional Recurrent Neural Networks/Long-Short Term Memory/Gate Recurrent Unit as its two ends. In this paper, we use
RNN and DCNN as our encoder. As for the decoder, we choose RNN as our decoder, which at each step through an attention mechanism combines the RNN and DCNN representations and outputs the corrected text.

2. Encoder And Decoder

2.1. RNN-DCNN Encoder

As for the encoder, we choose to combine RNN and DCNN for the representation of the erroneous OCR transcribed snippets at character level. During the decoder phase, RNN first corrects the errors one character at a time by employing an attention mechanism that combines the encoder representation. The encoder’s structure is as follows:

According to the RNN’s architecture[1],

\[ h(t) = f(h_{t-1}, x_t) \]  

(1)

Here, \( h_t \) denotes the current hidden state \( h \) at the time \( t \). \( x_t \) denotes the current input \( x \) at the time \( t \). It can be obviously seen that in RNN, current hidden state is determined by the current input \( x \) and the hidden state \( h_{t-1} \).

When we get the state of the hidden layer at all times, we can output the intermediate vector by the following function:

\[ C = q(h_1, h_2, h_3, h_T) \]  

(2)

To better improving the model’s encoding ability, we apply a Bidirectional RNN that reads the erroneous OCR snippet \( x_1, x_2, ..., x_T \) and encoding it into

\[ h_T = \left( \begin{array}{c} \overrightarrow{RNN} \\ \overleftarrow{RNN} \end{array} \right) \]  

(3)

Furthermore, we apply Deep Convolutional Neural Nets to capture the local context (i.e. compound information) of tokens. DCNN through their kernels limit the influence that characters beyond a token's context may have in determining whether the subsequent decoded characters forming a token should be split or merged[2]. The kernel size here is set to be 3. Due to the fact that every sequence input is decoded...
at character level, determining the right granularity of representation is not trivial. In between each of the layer, we apply non-linearity such as gated linear units to control the information filtered.

2.2. RNN-based Decoder
The process of decoding can be regarded as the reverse process of encoding. In this stage, we predict the next generated word by the following function:

\[
y_t = \arg \max P(y_t) = \prod_{t=1}^{T} p(y_t | \{y_1, y_2, \ldots, y_{t-1}\}, C)
\]  

(4)

As is clearly seen above, although encoder-decoder structure improve the ability of long series prediction to a certain extent, its ability will be degraded with the increase of input sequence. As can be seen from the above formulas, encoder aims to transform input information at all times into a vector of fixed length. With the increasing input scale, for example, when archaeologists find an archaeological data with extremely official content, this fixed length vector is not enough to provide enough information to express all input information.

To solve this problem, we thought of using attention mechanism[7]. Attention mechanism is a kind of mechanism widely used in sequence prediction model in recent years. When decoding, or generating model output, the mechanism will indicate which part of the input sequence the next output should focus on according to the attention range, and then output according to the region of interest, so as to reciprocate.

Attention mechanism greatly solves the limitations of traditional encoder-decoder model. When the decoder generates each token, it not only refers to the vector generated by the encoder and the current corresponding input, but also refers to the input text sequence within a certain range.

We can change the previous conditional probability formula to the following form:

\[
p(y_t | y_1, \ldots, y_{t-1}, X) = g(y_{t-1}, s_t, c_t)
\]

(5)

Here, \(s_t\) denotes the current hidden state. That means:

\[
s_t = f(s_{t-1}, y_{t-1}, c_t)
\]

(6)

It should be noticed that the conditional probability here is responding to the relevant content vector \(c_t\). However, in the traditional way, \(c_t\) is fixed. To fix that problem, we propose a new method, which is:

\[
c_t = \sum_{j=1}^{T} a_{ij} h_j
\]

(7)

Here, \(a_{ij}\) represents the magnitude of the influence received by the j-th input when the i-th output is generated. In order to normalize the final results, we use the softmax algorithm to calculate it.

\[
a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} \exp(e_{ik})}
\]

(8)

\[
e_{ij} = a(s_{t-1}, h_j)
\]

(9)

2.3. Training Strategy
Its good Conventionally, the cross-entropy loss function is widely used in encoder-decoder architectures for its excellent performance on the topic like classifying[8]. In this paper, we propose a new method to improve cross-entropy loss function’s performance in this task.

We propose a loss function that rewards higher models for their correcting behavior. The modified loss function is shown below.

\[
L_{\text{new}} = -(y \ln a + (1 - y) \ln(1 - a)(1 - \lambda P_{\text{src}} P_{\text{tgt}}))
\]

(10)
Here, if the input characters is exactly the same as the output character, which is our expectation, $P_{src}$ and $P_{arg}$ yields 1, otherwise 0. So it’s obvious that we can weight the value of cross-entropy function by the similarity of the input character and the output character. Our loss function rewards higher the model’s ability to correct erroneous sequences.

3. Results & Discussion

All post-hoc OCR correction approaches under comparison reduce significantly the amount of OCR errors. The Table below provide an overview of the performance as measured through Word Error Rate(WER) metrics[4].

| Model       | WEB   |
|-------------|-------|
| BiRNN       | 31.20 |
| Transformer | 8.11  |
| Seq2Seq     | 14.50 |
| Seq2Tree    | 11.20 |
| Our Model   | 6.97  |

Reducing the value of WER is significantly crucial for the task of Optimal Character Recognition task. It can be seen from the table that our model has achieved state-of-the-art(SOTA) in the historical datasets. This presents a relative decrease of $\delta = 82\%$.

For further study of the effectiveness of our new loss function based on cross-entropy loss, we try the different $\lambda$ to get the best hyper-parameters for our model. The results are as follows:

| Model $\lambda$ | WEB |
|-----------------|-----|
| $\lambda = 0.1$| 9.57|
| $\lambda = 0.2$| 9.44|
| $\lambda = 0.3$| 9.29|
| $\lambda = 0.4$| 9.03|
| $\lambda = 0.5$| 9.21|
| $\lambda = 0.6$| 9.38|

4. Conclusion And Future Work

In this work we propose a new model combined with BiRNN and DCNN towards post-hoc correction. Some popular feature extractors like Transformer[5] and Seq2Seq have limited utility in such dataset. However, by connecting BiRNN and DCNN together we can still get a good OCR effect even when the quality of the data set is extremely poor. Also, attention mechanism contributes a lot to our model’s decoding ability for it can provide a wider scope when decoding the currently generating token. Through our model, we achieve the great WER reduction rates with SOTA achievement.

The future work should mainly focus on the data augment. Nowadays, historical corpus tend to face the challenge like poor quality of scanning which varies greatly across books and pages and the absence of some crucial character in a successive sequence. Also, language evolution can exert a great challenge to us for variations of word spelling across centuries can easily lead to systematic correction mistakes.

Acknowledgment

This work was supported by the key project Science and Technology cooperation of Fujian province(no. 2017H007).
References

[1] Marwan Ali Albahar. “Recurrent Neural Network Model Based on a New Regularization Technique for Real-Time Intrusion Detection in SDN Environments”. In: Secur. Commun. Networks 2019(2019), 8939041:1–8939041:9. DOI : 10.1155/2019/8939041. URL : https://doi.org/10.1155/2019/8939041.

[2] Ian Colbert et al. “A Competitive Edge: Can FPGAs Beat GPUs at DCNN Inference Acceleration in Resource-Limited Edge Computing Applications?” In: CoRR abs/2102.00294 (2021). arXiv: 2102.00294. URL : https://arxiv.org/abs/2102.00294.

[3] Rui Dong and David Smith. “Multi-Input Attention for Unsupervised OCR Correction”. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers. Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, 2018, pp. 2363–2372. DOI : 10.18653/v1/P18-1220. URL : https://www.aclweb.org/anthology/P18-1220/.

[4] Jinxī Guo et al. “Efficient Minimum Word Error Rate Training of RNN-Transducer for End-to-End Speech Recognition”. In: Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020. Ed. by Helen Meng, Bo Xu, and Thomas Fang Zheng. ISCA, 2020, pp. 2807–2811. DOI : 10.21437/Interspeech.2020-1557. URL : https://doi.org/10.21437/Interspeech.2020-1557.

[5] Marius Mosbach et al. “On the Interplay Between Fine-tuning and Sentence-Level Probing for Linguistic Knowledge in Pre-Trained Transformers”. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020. Ed. by Trevor Cohn, Yulan He, and Yang Liu. Association for Computational Linguistics, 2020, pp. 2502–2516. DOI : 10.18653/v1/2020.findings-emnlp.227. URL : https://doi.org/10.18653/v1/2020.findings-emnlp.227.

[6] Yunsheng Shi et al. “A Normalized Encoder-Decoder Model for Abstractive Summarization Using Focal Loss”. In: Natural Language Processing and Chinese Computing - 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26-30, 2018, Proceedings, Part II. Ed. by Min Zhang et al. Vol. 11109. Lecture Notes in Computer Science. Springer, 2018, pp. 383–392. DOI:10.1007/978-3-319-99501-4_34.URL:https://doi.org/10.1007/978-3-319-99501-4_34.

[7] Ashish Vaswani et al. “Attention is All you Need”. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. Ed. by Isabelle Guyon et al. 2017. pp. 5998–6008. URL:https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

[8] Lin-Chen Weng, Ahmed M. Elsawah, and Kai-Tai Fang. “Cross-Entropy Loss for Recommending Efficient Fold-Over Technique”. In: J. Syst. Sci. Complex. 34.1 (2021), pp. 402–439. DOI : 10.1007/s11424-020-9267-9. URL : https://doi.org/10.1007/s11424-020-9267-9.