A Context-Free Spelling Correction Method for Classical Mongolian

Min Lu, Feilong Bao*, and Guanglai Gao

College of Computer Science, Inner Mongolia University, Hohhot, China
csfeilong@imu.edu.cn

Abstract. Spelling errors in the classical Mongolian text are mainly caused by misuse of polyphonic letters which present the same shape in the certain position of the word. About half to three-quarters of the classical Mongolian words are misspellings which have the correct appearances but wrong codes. In this paper, we code the Mongolian words by glyph codes to map the words to their shapes one-to-one. In addition, we also proposed the correction of out-of-vocabulary words (OOV) based on the Evolved Transformer by formalizing the correction task as a translation from misspellings to target spellings. The experimental results show that this approach achieves the new state-of-the-art performance.

Keywords: Classical Mongolian · Spelling errors · Glyph codes · OOV correction

1 Introduction

Spelling correction is the very first step of a natural language processing (NLP) system to ensure that all of the input text is spelled correctly before they were fed into subsequent steps. Most traditional systems in this field were built on Levenshtein Distance and statistical methods [16, 6, 19]. Recently, Recurrent Neural Networks (RNNs) have been found to be effective in natural language correction [20]. Chollampatt and Ng [2] proposed character-level statistical machine translation to "translate" unknown words into correct words. The nested RNN [7] were used to handle spelling correction in English which consists of a character-level RNN (CharRNN) and a word-level one (WordRNN). In addition, attention was further extended by Vaswani et al. [18], where the self-attentional Transformer architecture achieved state-of-the-art results in Machine Translation. Aiming to automate the laborious process of designing neural networks architectures, Evolved Transformer [14] was proposed then and achieved the more outstanding performance in Machine Translation. Transformers have not been applied to the spelling correction yet.

The existing methods, however, cannot be directly applied to classical Mongolian words. Classical Mongolian is an alphabetic writing language that uses

* Corresponding author: csfeilong@imu.edu.cn

[3] as the standard code. It is designed to encode every speech sound (vowel and consonant) with a Unicode code point. In classical Mongolian, a single encoded letter may map to many different glyphs, and a single glyph may map to many different encoded letters, too. For example, a letter "U+1820" maps to , , , and according to its contexts. On the other hand, a glyph maps to letter "U+1832" and letter "U+1833". The mismatch between encoded letters and glyphs makes classical Mongolian very easy to be misspelled. Numerous words present correct appearance but are typed with incorrect codes in the text since all the text editors care about is the appearance of the input words and nothing else. Those misspellings with correct shapes are formally known as pronunciation errors which we devote ourself to solve in this paper. They make up more than 90% in Mongolian spelling errors.

To solve the Mongolian spelling errors, an efficient mapping strategy between the word and shape is crucial before correction. [13] and [21] proposed a presentation character to represent the word based on morphology. However, it is a knowledge-dependent and labor-intensive work. In this paper, we propose the direct mapping between words and shapes by representing the words with glyph codes instead of Unicode codes. Then utilizing the vocabulary which is also coded in glyph code to find out the correct spelling. In addition, the Evolved Transformer [14] is applied to correct the out-of-vocabulary words (OOV). The glyph-code representation proposal successfully avoids complex mapping algorithm [13, 21]. It is also served for the generation of the training data for OOV correction module, which can be seen as the translation from glyph codes to their correct spellings. The main contributions are as follows:

- Glyph code is presented to convert the Mongolian text encoded with the standard code into its corresponding glyph representation. It not only avoids the costly resource collection process such as error patterns but makes words matched the dictionary efficiently.
- We apply the Evolved Transformer to correct OOV words. It yields significant improvements over the overall correction performance.

In the following sections, we introduce the related work firstly, then propose our solution, describe the experiments, and close the paper with a conclusion at last.

2 Related Work

Neural network models have not drawn much attention in spelling correction for classical Mongolian yet. The most successful spelling correction system is the dictionary-based system [13] which was the first work to use presentation character to represent the word. It mapped the Unicode code to presentation character by rules. Most works operated on Mongolian standard codes directly. The earliest rule-based MHAHP system [4] was just designed to detect Mongolian spelling errors. [5] extended the rules and summarized the common errors. However, the performance was limited by the complex error types. [15] proposed statistical translation framework, which regarded spelling proofreading as the translation from the wrong words to the correct ones.

Dictionary-based approaches typically perform poorly in the absence of complete vocabularies. More recent work in other languages has applied character-level models to machine translation and speech recognition as well, suggesting that it may be applicable to many other tasks that involve the problem of OOVs [8, 10, 1]. [20] present the character-based RNN model with an attention mechanism for performing language correction which allows for orthographic errors to be captured and avoids the OOV problem suffered by word-based neural machine translation methods. [12] demonstrate the use of LSTM with a delay, for jointly learning error patterns and language models for detection and correction in Indic OCR. [11] proposed semi-character RNN based on psycholinguistics experiments. More recently, the QANet [22] and Evolved Transformer [14] architectures alternate between self-attention layers and convolution layers for Question Answering applications and Machine Translation respectively.

The research on the correction of Mongolian out-of-vocabulary words (OOV) has not been sufficiently discussed yet. In this work, we utilize both the lexicon which maps the glyph codes of the words to their correct spellings and novel Evolved Transformer architecture to correct the pronunciation errors in Mongolian text.

3 Method

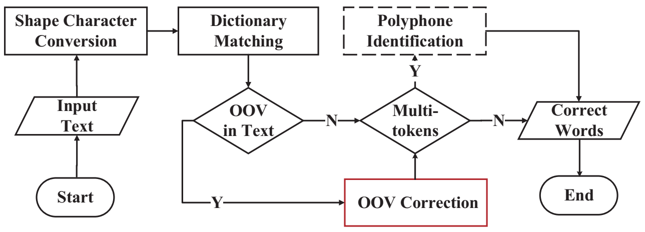

Fig. 1. General framework of Mongolian spelling correction.

Fig 1 shows the general framework of Mongolian spelling correction. Firstly, input strings are encoded into shape codes, then undergo dictionary matching to restore the words from shape representation. The dictionary is built manually with a vocabulary of 172,456, of which about 20,000 items are loan words. The key of each item is the Mongolian glyph codes, and the value is the correct spelling. We add the OOV correction module to the general architecture by applying novel seq2seq model – Evolved Transformer [14]. The polyphone error won't be discussed in this work for it belongs to context-sensitive real-word error. For convenience, words are presented with their national Latin transliteration (keyboard correspondence).

3.1 Glyph Code Conversion

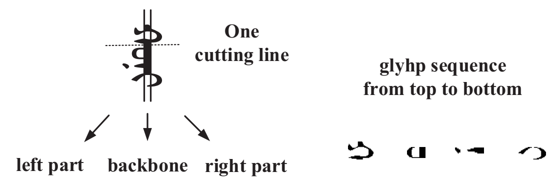

Fig. 2. The Mongolian word and its glyph code sequence.

Glyph Segmentation Process Glyph code encoding is to code the words in their glyph sequence in order after segmentation by the cutting line from top to bottom. As seen in Fig 2, the word ᠬᠣᠨᠢ (meaning: sheep, Latin: hqni) were segmented into 4 glyphs.

In this work, we adopt the simple and efficient segmentation algorithm inspired by the characteristic of the backbone. Classical Mongolian is written vertically. Each letter in a word is linked from top to bottom by a vertical line which is called the backbone of the word. The backbones are mostly placed in the middle of the words, consisting of dense black pixels with blank space on one or both sides. The projection line of the backbone on the horizontal x-axis serves as the cutting line. There is at most one or two consecutive white-pixel (255 in value) regions and one consecutive black-pixel (0 in value) region. The algorithm for determining the cutting line is as follows: First, being scanned from top to bottom, lines with only one continuous black pixels are chosen as the candidate cutting lines. Then the frequency of each candidate and its numbers of the dependent lines[1] are recorded. After that, each candidate is scored by the summation of its own frequency and its number of the dependent line. Finally, the one with the highest score is identified as the cutting line.

100 glyphs are defined in total after segmentation of about 200,000 distinct words (See Fig 3). The images were all generated as binary images in Portable Network Graphic Format (.png). A glyph can be composed of multiple letters, and a letter can be composed of several glyphs as well. The glyphs with higher labels such as glyph 93, can be further segmented by multi-level segmentation. In this work, only one-level segmentation is considered which can perform well enough.

Glyph Segmentation Reasoning Why not use the entire word image directly to represent the words instead of their glyph sequence? It's because some typeface differences are hard to distinguish with the naked eye. Taking the words

[1] If the black-pixel area of the line and other ones between them can completely cover the black pixels of the current candidate, the line is considered as the dependent line of the current candidate.

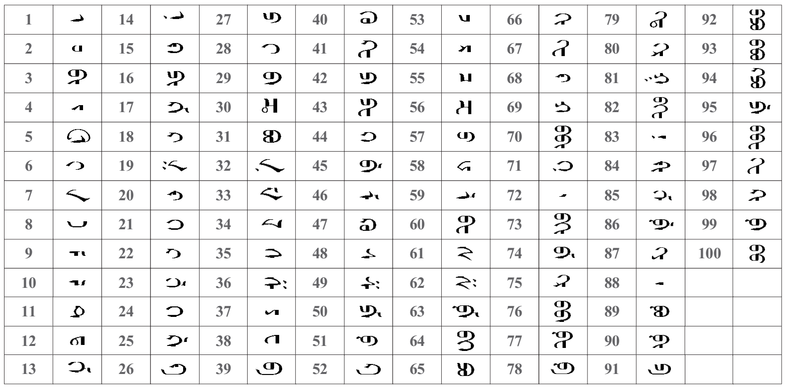

Fig. 3. Glyphs and its labels.

(Latin: aNharvn) and (Latin: aegharvn) for instance, the shape of the second letter is slightly different. The former is correctly coded by “U+1829” (Latin: N) and the latter is coded by two codes “U+1821” (Latin: e), “U+182D” (Latin: g) wrongly. These seemingly identical words mislead the user’s incorrect input. When we generate the entire word image for those words and match the dictionary (matching the pixel value), the matching will fail. In order to avoid the failure of in-vocabulary words correction caused by inappropriate matching algorithm, glyph sequence is adopted to represent the words. The most similar glyphs were categorized into the same groups. The words with a tiny difference in shape can match the same code sequences with the target words so that they can be matched successfully.

3.2 OOV Correction

We choose to use the Evolved Transformer architecture [14] to “translate” OOV misspellings into correct words. The input of the model is the corresponding labels of glyphs as seen in Fig 3. The output is the correct Unicode spelling. It produces a ranked list of candidate words, in which the top one is the most likely result. As for the word (See Fig 2), the input sequence should be the array of [69, 2, 83, 6], and the output is the letters of correct spelling [“U+182C”, “U+1823”, “U+1828”, “U+1822”] (Latin: [h, q, n, i]).

The goal of the Evolved Transformer is to examine the use of neural architecture search methods to design better feed-forward architectures for seq2seq tasks. It applies tournament selection architecture search and warm start it with the Transformer to evolve the better and more efficient architecture. The encoding search space is inspired by the NASNet search space [9], but is altered to allow it to express architecture characteristics found in recent state-of-the-art feed-forward seq2seq networks. The search space was designed as one that can represent the Transformer with which the initial population was seeded. It consists of two stackable cells, one for the model encoder and one for the decoder

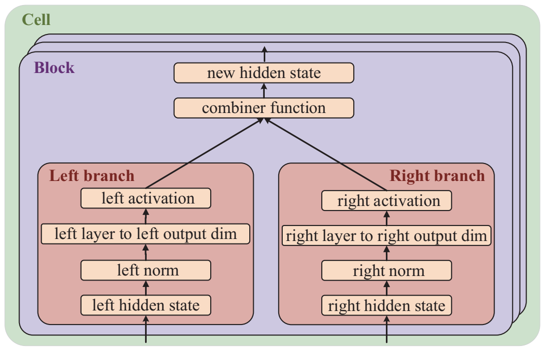

Fig. 4. Architecture composition from encoding.

(see Fig 4.). Each cell contains NASNet-style blocks, which receive two hidden state inputs and produce new hidden states as outputs. The encoder contains six blocks and the decoder contains eight blocks, so that the Transformer can be represented exactly. The blocks perform separate transformations to each input and then combine the transformation outputs together to produce a single block output; The transformations applied to each input is referred to as a branch. The search space contains five branch-level search fields (input, normalization, layer, output dimension and activation), one block-level search field (combiner function) and one cell level search field (number of cells).

To perform the search directly on the computationally demanding task, Progressive Dynamic Hurdles (PDH) was developed to dynamically allocate more resources to more promising candidate models. The model can learn morphological knowledge well to find and organize the proper letters corresponding to the glyph codes. In the absence of a dictionary as a reference, it is adapted to apply for solving spelling errors.

4 Experiment

In this section, we will first evaluate the performance of OOV correction by comparing the Evolved Transformer to the Transformer model. All the comparative experiments were conducted in the Tensor2Tensor [17] training regime on a single GPU. OOV correction is intended to generate words which conform to Mongolian morphological rules according to input word shapes. We evaluate it by shape accuracy and spelling accuracy. Shape accuracy measures the performance of generating words which can present the same shape with the input words without caring about the spellings. The latter one, as it implies, evaluates the performance of generating the correct spellings. They were all tested on common words and loan words. We will then benchmark the performance of our overall performance against the evolutionary method [13]. It is evaluated by the standard binary classification evaluation metrics of *precision*, *recall* and *f*-measure.

4.1 Experimental Data

The datasets for the Evolved Transformer is the aligned sets of glyph sequences and correct spellings extracted from the dictionary. The training set consisted of 140,000 items including loan words of about 15,000 items and 125,000 items of commonly used words. 6,000 items from the remaining entries of the dictionary were extracted as test data for OOV correction. Loan words and commonly used words were accounted for half of the set respectively. For testing the overall performance, we firstly crawled texts of 35MB from the Mongolian news website. The words whose frequencies were not too high or too low were selected Then. After filtering shape errors among them, we selected the 8,000 tokens at last to correct them manually by several native Mongolian people. The aligned set of with and without correction is used to test the overall performance.

4.2 Result and Analysis

Table 1. Comparison between the Transformer and ET on different models.

Model	Model Size	Common Set		Loan Set	
		Spelling Acc	Shape Acc	Spelling Acc	Shape Acc
ET	Tiny	96.40%	99.83%	82.16%	96.70%
Transformer	Tiny	93.70%	99.56%	76.16%	95.67%
ET	Base	84.03%	98.53%	67.90%	90.80%
Transformer	Base	81.96%	97.16%	66.10%	89.23%

OOV Words Correction As we can see in Table 1 and Table 2, we compared the Evolved Transformer (ET) to the Transformer on variant model scales and embedding sizes. Seeing from the results, ET demonstrates stronger performance than the Transformer almost at all sizes, with the largest difference of 6.00% in spelling accuracy and 1.57% in shape accuracy. By the "Tiny" ET, we get the comfortable results that the highest spelling accuracy of commonly-used words reaches 96.40% and that of loan words reaches 82.16%. The shape accuracy is generally higher and more stable than spelling accuracy with the highest accuracy of 99.83%.

As demonstrated in Table 1, we choose the smaller models as "Tiny" and "Base" size models for our character-level task of which the vocabulary is only 100. The result indicates that the "Tiny" model fits our task better. "Tiny" model yields the better results both in the correction of commonly used words and loan words than the "Base" model. It's mainly because of the degeneracy. Beyond that, for both the Transformer and ET, we tried three additional embedding sizes [64, 256, 384] under the better architecture – "Tiny" size model to explore the best embedding size for our task. As can be seen in Table 2, the embedding size

Table 2. Comparison between the Transformer and ET in different embedding size.

Model	Embedding Size	Common Set		Loan Set	
		Spelling Acc	Shape Acc	Spelling Acc	Shape Acc
ET	64	96.10%	99.60%	80.06%	96.50%
Transformer	64	92.16%	99.60%	74.53%	95.10%
ET	128	96.40%	99.83%	82.16%	96.70%
Transformer	128	93.70%	99.56%	76.16%	95.67%
ET	256	96.30%	99.60%	80.06%	96.43%
Transformer	256	95.80%	99.36%	79.06%	96.50%
ET	384	83.80%	98.36%	67.46%	89.93%
Transformer	384	95.60%	99.50%	79.46%	96.33%

larger than 128 did not make any improvement for ET. While the performance of the Transformer continued to grow until the size 384. The gap between the two models becomes smaller with the increasing of embedding size. With the consideration of both model size and effectiveness, the "Tiny" model at its default setting is identified as our method for correcting OOV words.

It should be noted that the correction performance of loan words is far lower than commonly-used words. It's mainly for inadequate loan words in the training data. The model failed to learn the specific word-formation rules of loan words and usage of control character[2], which is written frequently in loan words but hardly in common words.

Overall Performance We conduct three experiments to show respectively the quality of the construction of our dictionary and the correction effect of OOV words as shown in Table 3. The baseline model [13] is the superior work based on rules and dictionaries. From the experiment, we can see that the proposed methods with and without OOV correction module are all outperforms the baseline systems in each criteria. The f-measure of the proposed method is respectively 3.30% and 6.39% higher than the baseline model. It can also be seen that the recall rate is relatively lower in both single dictionary matching methods. It is a common shortcoming in every dictionary-based system because the dictionary cannot cover all the human words, especially for languages with rich inflection. After joining the OOV correction module, the recall is improved further by 5.99% against the single glyph match method. It leads to a 3.00% increase in the f-measure finally.

The experiment shows the excellent results, which confirms that our dictionaries, as well as the OOV correction module, were built pretty well.

[2] Control characters are used in conjunction with Mongolian letters to control the word shapes. They mainly refer to three Mongolian Free Variation Selector: "U+180B", "U+180C", "U+180D".

Table 3. Evaluation for the overall performance. G_Match refers to the performance after the glyph-based dictionary matching only. G_Match+ET refers to the experimental effect of adding OOV correction by the Evolved Transformer.

Criteria	Baseline	G_Match	G_Match+ET
P	95.13%	99.03%	99.04%
R	91.00%	93.31%	98.90%
F	93.02%	96.09%	98.97%

5 Conclusion

To address the serious pronunciation errors in classical Mongolian text, this paper proposed the glyph codes to represent the words. It is an efficient representation method which avoids designing and building enormous and complicated mapping rules. In addition, we proposed the first OOV correction module and add it to the general correction framework. It is handled by the Evolved Transformer model with great performance. Experimental results demonstrate that our method can meet the practical demands well. In future work, we will expand the training data further for loan words and will also explore a uniform architecture to correct both OOV words and polyphones.

6 Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61563040, 61773224); Natural Science Foundation of Inner Mongolia (Nos. 2018MS06006 2016ZD06).

References

1. Chan, W., Jaitly, N., Le, Q., Vinyals, O.: Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4960–4964. IEEE (2016)
2. Chollampatt, S., Ng, H.T.: Connecting the dots: Towards human-level grammatical error correction. In: Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications. pp. 327–333 (2017)
3. GB25914-2010: Information technology of traditional Mongolian nominal characters, presentation characters and control characters using the rules. China National Standardization Technical Committee, Beijing (2010)
4. Hua, S.: Modern mongolian automatic proofreading system—mhahp. Journal of Inner Mongolia University:Philosophy and Social Science Edition (4), 49–53 (1997)
5. Jiang, B.: Research on Rule-Based method of Mongolian automatic correction. Ph.D. thesis (2014)
6. Kernighan, M.D., Church, K.W., Gale, W.A.: A spelling correction program based on a noisy channel model. In: Proceedings of the 13th conference on Computational linguistics-Volume 2. pp. 205–210. Association for Computational Linguistics (1990)

7. Li, H., Wang, Y., Liu, X., Sheng, Z., Wei, S.: Spelling error correction using a nested rnn model and pseudo training data. arXiv preprint arXiv:1811.00238 (2018)
8. Ling, W., Trancoso, I., Dyer, C., Black, A.W.: Character-based neural machine translation. arXiv preprint arXiv:1511.04586 (2015)
9. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 19–34 (2018)
10. Maas, A., Xie, Z., Jurafsky, D., Ng, A.: Lexicon-free conversational speech recognition with neural networks. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 345–354 (2015)
11. Sakaguchi, K., Duh, K., Post, M., Van Durme, B.: Robsut wrod reocginiton via semi-character recurrent neural network. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
12. Saluja, R., Adiga, D., Chaudhuri, P., Ramakrishnan, G., Carman, M.: Error detection and corrections in indic ocr using lstms. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 17–22. IEEE (2017)
13. Si, L.: Mongolian proofreading algorithm based on non-deterministic finite automata. Journal of chinese information processing **23**(6), 110–6 (2009)
14. So, D.R., Liang, C., Le, Q.V.: The evolved transformer. arXiv preprint arXiv:1901.11117 (2019)
15. Su, C., Hou, H., Yang, P., Yuan, H.: Based on the statistical translation framework of the mongolian automatic spelling correction method. J. Chin. Inf. Proces pp. 175–179 (2013)
16. Toutanova, K., Moore, R.C.: Pronunciation modeling for improved spelling correction (2002)
17. Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A.N., Gouws, S., Jones, L., Kaiser, L., Kalchbrenner, N., Parmar, N., Sepassi, R., Shazeer, N., Uszkoreit, J.: Tensor2tensor for neural machine translation. CoRR **abs/1803.07416** (2018), `http://arxiv.org/abs/1803.07416`
18. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
19. Wilcox-O'Hearn, A., Hirst, G., Budanitsky, A.: Real-word spelling correction with trigrams: A reconsideration of the mays, damerau, and mercer model. In: International conference on intelligent text processing and computational linguistics. pp. 605–616. Springer (2008)
20. Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., Ng, A.Y.: Neural language correction with character-based attention. arXiv preprint arXiv:1603.09727 (2016)
21. Yan, X., Bao, F., Wei, H., Su, X.: A novel approach to improve the mongolian language model using intermediate characters. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 103–113. Springer (2016)
22. Yu, A.W., Dohan, D., Luong, M.T., Zhao, R., Chen, K., Norouzi, M., Le, Q.V.: Qanet: Combining local convolution with global self-attention for reading comprehension. arXiv preprint arXiv:1804.09541 (2018)