CHINESE SPELLING TEXT GENERATION OF MATHEMATICAL FORMULAS

Su Dong[1], *Shan Liu*[2], *Sicen Liu*[1], *and Buzhou Tang*[1†]

[1] Harbin Institute of Technology, China
[2] Tencent Technology Co., Ltd. China
listenersu@gmail.com, shiningliu@tencent.com, liusicen@stu.hit.edu.cn, tangbuzhou@gmail.com

ABSTRACT

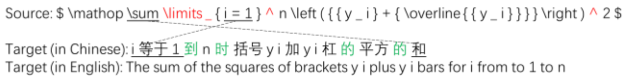
Recently, speech assistants have brought convenience to our lives from many aspects. In the education field, speech assistants can also help teachers to reduce their burdens. However, there is no suitable solution to synthesize speeches for mathematical formulas although there have been lots of good techniques for text-to-speech (TTS) in the general domain. One possible solution is that we can convert mathematical formulas expressed in the LaTeX format to spelling texts and synthesize them into speech. In this paper, we investigated text generation methods that translate mathematical formulas in LaTex into Chinese spelling texts. For this purpose, we first constructed a parallel corpus of mathematical formulas and Chinese spelling texts, then compared the existing commonly used text generation methods, such as rule-based, Seq2Seq, Transformer and Graph2Seq, and finally proposed a novel model. As far as we know, this is the first study for Chinese spelling text generation of mathematical formulas. Experiment results on the annotated corpus show that our proposed model significantly outperforms the existing commonly used generation models.

Index Terms—Mathematical Formals to Chinese spelling texts, text generation, text-to-speech

1. INTRODUCTION

Nowadays, text-to-speech (TTS) has been widely used in lots of application systems in many domains, such as speech assistants in our daily interactions. In the education field, online education has been becoming more and more popular, and the TTS systems can help teachers to reduce their burdens on courseware recording. However, in the case of mathematics teaching, we cannot directly synthesize speeches for mathematical formulas as mathematical formulas are usually stored in structured data format like LaTex. One possible solution to mathematical formulas to spelling texts is that we first generate spelling texts from mathematical formulas in the LaTeX format and then convert spelling texts into speeches.

Converting mathematical formulas in Latex into spelling texts (LaTeX-to-Text) is a new text generation problem, similar to Seq-to-Seq (Seq2Seq) or SQL-to-Text. There have been a large number of studies on Seq2Seq and SQL-to-Text problems.



```
Source: $ \mathop \sum \limits _ { i = 1 } ^ n \left ( { { y _ i } + { \overline{ { y _ i } } } } \right ) ^ 2 $
Target (in Chinese): i 等于 1 到 n 时 括号 y i 加 y i 杠 的 平方 的 和
Target (in English): The sum of the squares of brackets y i plus y i bars for i from to 1 to n
```

Fig. 1. An example of LaTex-to-ChineseText.

The early Seq2Seq model is proposed for machine translation [1], which usually uses Recurrent Neural Network(RNN) Encoder-Decoder framework with attention mechanism to find a target sentence that maximizes the conditional probability of a given source sentence. Minh, etc. [25] use local and global attention to guide the RNN-based Seq2Seq model to focus on not only the whole source text but also the local specific source fragment. Pointer Generator Network [24] is an architecture that extends the standard Seq2Seq model with attention mechanism by introducing the copy mechanism. The Graph2Seq model is a typical model designed for SQL-to-Text [23], which represents each SQL query as a graph instead of a sequence and uses Graph Neural Network to model SQL query.

Compared to the common text generation problems (Seq2Seq, SQL-to-Text, etc.), LaTex-to-Text suffers from the following three challenges, taking Chinese spelling text generation (called LaTex-to-ChineseText) for example:

(1) Ambiguity. The spelling texts of some mathematical symbols are ambiguous. For example, "+" can be spelled as "加"(plus) or "正"(positive), and "(1,2)" can be spelled as a coordinate of a point or an interval.

(2) Complex mapping. Some mathematical symbols correspond to complex spelling texts, such as the red symbols in Fig. 1, which are mapped to the Chinese characters in green. In this example, there are long-distance relationships among different symbols.

(3) Order inconsistency. The order of symbols in mathematical formulas is different from corresponding spelling texts. Fig. 1 gives an example, where the underlined symbol "\Sum" is before "i = 1", but the corresponding spelling text "和 (sum)" of "\Sum" is after "i 等于 1 (i from 1)" of "i = 1".

In this paper, we investigated the LaTeX-to-ChineseText problem. We first constructed a parallel corpus,

†Corresponding Author

978-1-6654-0540-9/22/$31.00 ©2022 IEEE

then compared different methods commonly used for text generation, and finally proposed a novel model that considers the characteristics of LaText-to-ChineseText. Our proposed model is a Seq2Seq model incorporating a dictionary of mappings from mathematical formula symbols to Chinese vocabularies.

The main contributions of our work can be summarized as follows:

–We built the first corpus and several baselines for the LaText-to-ChineseText.

– We propose a novel generation model to convert mathematical formulas in the LaTeX format to human-readable Chinese spelling texts. This model significantly outperforms other popular models for text generation.

2. RELATED WORKS

For text generation, Seq2Seq models [1, 2] are classic and powerful models. They have achieved great performance in many applications such as machine translation. These models even exhibit human-level performance in certain important domains [3, 4]. A Seq2Seq model is composed of two modules: encoder and decoder. The encoder is responsible for encoding the source language sequence, and the decoder is responsible for decoding the semantic vector outputted by the encoder into the target language sequence. The basic Seq2Seq model usually uses the recurrent neural network (RNN) [5] as encoder and decoder. Sometimes other neural networks are also used as encoder or decoder, such as long short-term memory network (LSTM) [6], gated recurrent unit (GRU) [7], convolutional neural network (CNN) [8, 9], etc.

Transformer [10] is a model that learns the dependencies between words based entirely on self-attention without any recurrent or convolutional layers. It has achieved state-of-the-art results on the machine translation task. In recent years, many variants of Transformer such as GPT [11], BERT [12], Transformer-XL [13], Universal Transformer [14], Star-Transformer [15], and CN3 [16], have also been proposed.

Graph2Seq is a graph neural network-based model for text generation, where the graph neural network is used as encoder for graph input and the same decoder as the Seq2Seq model is adopted. In graph neural network, the information fusion progresses via message-passing across the whole graph. The common graph neural networks include graph convolutional networks (GCN) [17], graph attention network (GTA) [18], gated graph neural network (GGNN) [19, 20] and so on. In our experiment, we convert mathematical formulas in LaTeX into hierarchical trees, and use GCN to encode the formula trees.

3. METHODOLOGY

The overview architecture of our proposed model is shown in Fig. 2. The proposed model uses the basic Seq2Seq model as backbone and introduces a dictionary of mappings from mathematical formula symbols to Chinese vocabularies to guide Chinese spelling text generation in decoder.

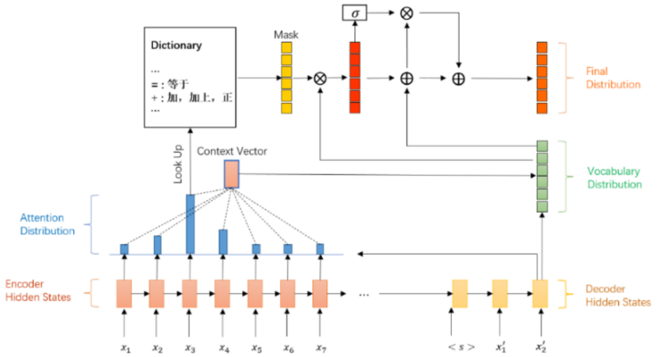


Fig. 2. Over architecture of our proposed model, that is, a Seq2Seq model incorporating a dictionary of mappings from mathematical formula symbols to Chinese vocabularies.

3.1. Mapping Dictionary Construction

According to the statistics on our annotated corpus, we found that more than 60% mathematical formula symbols have one or more clear spelling texts. Therefore, it is straightforward to construct a dictionary mapping from mathematical formula symbols to Chinese vocabularies to guide Chinese spelling text generation. Table 1 list some examples of the mappings.

Mathematical formula symbols	Chinese spelling texts
+	加 (add/plus), 加上 (add/plus), 正 (plus/positive)
-	减 (minus), 减去 (minus), 负 (minus/negative)
±	加减 (add and subtract), 正负 (plus-minus)
(	括号 (bracket), 开区间 (open interval), 左开区间 (left open interval)
=	等于 (equal)
$\backslash ge$	大于等于 (greater or equal)
$\backslash sqrt$	根号 (sqrt)

Table 1. Examples of mappings. Symbols "+", "-", "±", and "(" correspond to multiple Chinese spelling texts, while symbols "=", "$\backslash ge$", and "$\backslash sqrt$" only correspond to one Chinese spelling text.

3.2. Seq2Seq with Mapping Dictionary

How to use the mapping dictionary to guide the model generating Chinese spelling texts is another problem. Our idea is that at each time step in the decoding process, for a mathematical formula symbol in the mapping dictionary, the corresponding spelling texts should be generated with relatively high probabilities. The detailed information of our proposed model is presented as below.

Similar to [21], after a sequence of the mathematical formula symbols x_i are sent to the RNN encoder, a sequence of encoder hidden states h_i is generated. In the decoding process, the decoder hidden state s_t will also be generated at each time step t. We use the attention mechanism to find the most relevant mathematical formula symbols at each time step in the decoding process. The attention score a^t is calculated as in [1]:

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn}), \quad (1)$$

$$a^t = softmax(e^t), \quad (2)$$

$$h_t^* = \sum_i a_i^t h_i, \quad (3)$$

where v, W_h, W_s and b_{attn} are learnable parameters, and h_t^* is the context vector produced by a weighted sum of the encoder hidden states. The attention score can be viewed as a distribution over the source mathematical formula symbols at each decoding time step. Therefore, the most relevant mathematical formula symbol's index m at each time step is calculated as follows:

$$m = \operatorname*{argmax}_i(a_i^t) \quad (4)$$

Then, we can get the most relevant mathematical formula symbol x_m^t at each decoding time step from the mathematical formula symbols sequence.

To take advantage of the mapping dictionary constructed in the above section, we introduce a mask vector P_{mask} based on x_m^t and the mappings dictionary. P_{mask} has the same dimensions as the vocabulary distribution vector P_{vocab}. If x_m^t is in the mapping dictionary, then we get the Chinese spelling texts corresponding to x_m^t and set those texts' values in corresponding dimensions of P_{mask} to 1, and the other dimensions' values of P_{mask} are set to 0. Otherwise, the mask vector P_{mask} is set to an all-zero vector. The possibility distribution of Chinese spelling texts for a formula symbol based on the mapping dictionary P_{rel} can be calculated as follows:

$$P_{vocab} = softmax(V'(V[s_t, h_t^*] + b) + b'), \quad (5)$$

$$P_{rel} = P_{mask} \odot P_{vocab}, \quad (6)$$

where V, V', b, and b' are learnable parameters and $\odot$ is the Hadamard product function.

Finally, we design a gating mechanism similar to that used in LSTM with a residual structure to calculate the final generated text distribution P_{final}:

$$P_{final} = softmax(\sigma(W_{rel} P_{rel} + b_{rel}) \odot P_{vocab} + P_{vocab}), \quad (7)$$

where W_{rel} and b_{rel} are learnable parameters, and σ is the sigmoid function.

4. EXPERIMENTS

4.1. Dataset

To construct the parallel corpus, we collected 138,125 high school math problems, and extracted 38,270 mathematical formulas for annotation. Two annotators were recruited for annotation and an education expert was recruited for judgment when there were disagreement between the two annotators.

The statistics of the annotated corpus are listed in Table 2. It can be seen that the ratio of mathematical formula symbols with multiple spelling texts and with complex spelling texts exceeds 50%. As mentioned above, this does cause certain difficulties in Chinese spelling text generation.

Statistical item	Value
The size of the dataset	38270
The average length of mathematical formulas	30.94
The average length of Chinese spelling texts	14.64
Number of mathematical formula symbols	281
Number of mathematical formulas symbols complex spelling text mappings	80
The ratio of mathematical formula symbols without clear spelling texts	32.33%
The ratio of mathematical formula symbols with one clear spelling texts	42.72%
The ratio of mathematical formula symbols with multiple clear spelling texts	24.95%

Table 2. Dataset statistics.

4.2. Baselines

Since there was no work to refer to this task, we chose popular text generation models as baselines. The baseline models include a rule-based method, two Seq2Seq models based on RNN, a Transformer-based model, and a Graph2Seq model.

In the Seq2Seq models, we used Bi-LSTM as encoder and LSTM as decoder with the attention mechanism. Among the two Seq2Seq models, one used the copy mechanism for the context of mathematical formulas, called Seq2Seq+copy. In the Graph2Seq model, we converted the formulas in LaTex into hierarchical trees by simple rules. Specifically, the mathematical formula symbols before "{ }" are regarded as parent nodes, and the symbols in "{ }" are regarded as child nodes. According to these rules, a mathematical formula can be converted into a tree, and then we used GCN to encode the formula tree. Fig. 3. gives an example of this conversion.

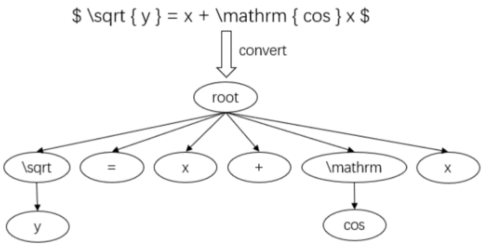


Fig. 3. Example of conversion from a mathematical formula to a formula tree.

4.3. Settings and Results

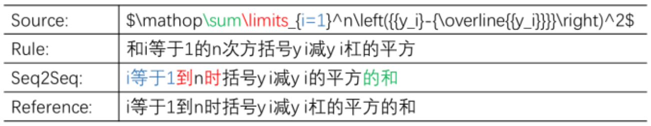
For all models, the dimensions of word embedings and hidden states were set to 256. During model training, all word embeddings were randomly initialized from a unified distribution, the Adam algorithm was selected as the optimizer, the batch size was set to 64, the dropout rate was set to 0.3, and the number of epochs was set to 100. At test time, we used beam search with beam size of 5 for Chinese spelling text generation. The annotated corpus was split into a training set, a development set and a test set at ratio of 8:1:1. The performances of all models were measured by BLEU-4 score. The results are shown in Table 3.

It can be seen that all neural network-based models are much better than the rule-based method except the Transformer-based model. The BLEU-4 score difference between the rule-based method and other models except Transformer exceeds 12 at least. The reason may be that the labeled datat is not enough to train a Transformer model. Compared to Graph2Seq, the two Seq2Seq models shows much better performance by near 20 in BLEU-4 score. In the case of the two Seq2Seq models, Seq2Seq+copy even shows a little worse performance, which indicates that the context of mathematical formulas has little effect on LaTex-to-ChineseText. When introducing the mapping dictionary, the new Seq2Seq model achieves the highest BLEU-4 score of 76.09. The performance gain from the mapping dictionary is 1.38 in BLEU-4 score. This result proves the effectiveness of our proposed model.

Model	BLEU-4
Rule	43.63
Transformer	42.37
Graph2Seq	55.65
Seq2Seq	74.71
Seq2Seq+copy	73.83
Our Model	**76.09**

Table 3. Results of different generation models.

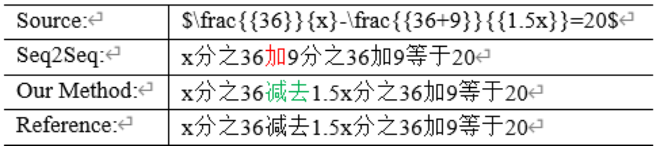
To verify the effectiveness of Seq2Seq models, we conducted a case study as shown in Fig. 4. It is clear that the Seq2Seq models can tackle some complex mapping and order inconsistency cases as mentioned in Fig. 1.



Source:	`$\mathop\sum\limits_{i=1}^n\left({{y_i}-{\overline{{y_i}}}}\right)^2$`
Rule:	和i等于1的n次方括号y i减y i杠的平方
Seq2Seq:	i等于1到n时括号y i减y i的平方的和
Reference:	i等于1到n时括号y i减y i杠的平方的和

Fig. 4. Case study for the Seq2Seq models.

To further verify the effectiveness of our models, we also conducted a case study as shown in Fig.5. It can be seen that the mathematical formula symbol "-" is incorrectly predicted by the Seq2Seq model to "加 (add/plus)", whose corresponding mathematical symbol is '+'. This may be because that the usage and context of '+' and '-' are similar. However, our proposed model corrects the error due to the mapping dictionary, where the possible Chinese spelling texts of '-' is limited to "减 (minus)", "减去 (minus)" or "负 (minus/negative)".



Source:↵	`$\frac{{36}}{x}-\frac{{36+9}}{{1.5x}}=20$`↵
Seq2Seq:↵	x分之36加9分之36加9等于20↵
Our Method:↵	x分之36减去1.5x分之36加9等于20↵
Reference:↵	x分之36减去1.5x分之36加9等于20↵

Fig. 5. Case study for our proposed model.

Although our proposed model shows much better performance than the other models, it also has some limitations. One limitation is that we did not consider complex mappings when constructing mapping dictionary. Another limitation is that all word embeddings were randomly initialized as there is no publically available pre-trained word embeddings for LaTex-to-ChineseText. The two directions for further improvements are: 1) adding complex mappings into mapping dictionary, and 2) training word embeddings on large-scale unlabeled data to replace the randomly initialized word embeddings.

5. CONCLUSION

In this paper, we investigated a new problem, that is, Chinese spelling text generation of mathematical formulas from corpus construction and methods. A parallel corpus of 38,270 mathematical formulas and corresponding Chinese spelling texts was constructed. On this corpus, we compared different types of methods, including the rule-based method, the Transformer-based model, and the Seq2Seq models. Furthermore, we proposed a novel Seq2Seq-based model. The proposed model outperforms other methods.

6. REFERENCES

[1] Bahdanau, Dzmitry, Kyung Hyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *3rd International Conference on Learning Representations, ICLR 2015*. 2015.

[2] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems*. 2014.

[3] Hassan, Hany, et al. "Achieving human parity on automatic Chinese to English news translation." *arXiv preprint arXiv:1803.05567* (2018).

[4] Yonghui, W., et al. "Bridging the gap between human and machine translation." *arXiv preprint arXiv:1609.08144* (2016).

[5] Elman, Jeffrey L. "Finding structure in time." *Cognitive science* 14.2 (1990): 179-211.

[6] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[7] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *NIPS 2014 Workshop on Deep Learning, December 2014*. 2014.

[8] Gehring, Jonas, et al. "A Convolutional Encoder Model for Neural Machine Translation." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

[9] Gehring, Jonas, et al. "Convolutional sequence to sequence learning." *International Conference on Machine Learning*. PMLR, 2017.

[10] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017.

[11] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[12] Devlin, Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *NAACL-HLT (1)*. 2019.

[13] Dai, Zihang, et al. "Transformer-XL: Attentive Language Models beyond a Fixed-Length Context." *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019.

[14] Dehghani, Mostafa, et al. "Universal Transformers." *International Conference on Learning Representations*. 2018.

[15] Guo, Qipeng, et al. "Star-Transformer." *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019.

[16] Liu, Pengfei, et al. "Contextualized non-local neural networks for sequence learning." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. No. 01. 2019.

[17] Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." *arXiv preprint arXiv:1609.02907* (2016).

[18] Velickovic, Petar, et al. "GRAPH ATTENTION NETWORKS." *stat* 1050 (2018): 4.

[19] Li, Yujia, et al. "Gated Graph Sequence Neural Networks." (2017).

[20] Beck, Daniel, Gholamreza Haffari, and Trevor Cohn. "Graph-to-Sequence Learning using Gated Graph Neural Networks." *arXiv e-prints* (2018): arXiv-1806.

[21] Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." *arXiv preprint arXiv:1602.06023* (2016).

[22] Xu, Kun, et al. "SQL-to-Text Generation with Graph-to-Sequence Model." *EMNLP*. 2018.

[23] See, Abigail, Peter J. Liu, and Christopher D. Manning. "Get To The Point: Summarization with Pointer-Generator Networks." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

[24] Luong, Thang, Hieu Pham, and Christopher D. Manning. "Effective Approaches to Attention-based Neural Machine Translation." *EMNLP*. 2015.