Chinese Math Word Problems Generation Network

Junjian Huang¹, Ming Xu², *, Hengyi Zheng¹ and Qian Shang¹

¹ College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
² Shenzhen University Information Center, Shenzhen, China

*xuming@szu.edu.cn

Abstract. Aiming at the feature vector bottleneck problem and the high time cost of the training process in the automatic generation of Chinese math word problems under the end-to-end architecture, we proposed an automatic generation method of Chinese math word problems based on the pre-training model combined with the integration of encoder and decoder. We used a deep neural network to model the mathematical equation sequence and Chinese keyword information, and used the stepped attention matrix to generate word problems. For training and testing on the Ape210K data set, compared with the end-to-end method, the Rouge-1 and Rouge-L evaluation indicators in our method was increased by 14.1% and 12.5%, as well as the training time cost was reduced by nearly 50%.

1. Introduction

Math word problems are an important part of mathematics education. They help students better understand and use mathematics knowledge through the scenes of life. An example of a Chinese math word problem is shown in Table 1. Under the background that the education industry is receiving more and more attention, personalized and customized mathematics education (including the customized generation of mathematics word problems) has a larger prospect.

At present, there are not many researches on the automatic generation of math word problems. The existing researches are generally classified as natural language processing text generation tasks. Nowadays, the task of text generation is mainly to use end-to-end networks, such as sequence-to-sequence model [1], attention mechanism [2]. Even with these methods, the generation of mathematical word problems is still a challenging task. There are 3 typical difficulties as following:

• Different from general text generation tasks (machine translation, etc.), the text of mathematical word problems not only need to be coherent, readable, have semantic information, and meet the desired story background, but also need to have logic between numbers, both of which are required considered at the same time.

• In the task of text generation, there are some hidden background knowledge that model may not be mastered. This is especially true when it comes to generating mathematical problems. Because the background knowledge of mathematical word problems is usually associated with numbers. It needs some derivation. For example, when generating the chicken and rabbit question, it is necessary to know that the chicken has two legs and the rabbit has four legs to correctly generate this type of problem.

• When using an end-to-end network, the intermediate vector representation between the encoder and the decoder may be the bottleneck of the network. A fixed-dimensional vector is
not enough to represent rich numbers and equation information because this could cause
information loss. At the same time, the training and prediction speed of the recurrent neural
network is not optimistic. Therefore, the cost is relatively high on large-scale data sets.

In order to tackle the above difficulties, we proposed a Chinese Math word problem GeNeration
Network(short for CMGNN). In our proposed model, 1) we preprocessed the equation by constructing
it into a Reverse Polish notation and encoding it with the keyword, so as to enhance the association
between the equation information and the keyword information. 2) We used a pre-trained language
model to help the model acquire hidden background knowledge. 3) We made the encoder and decoder
share the same network structure to solve the bottleneck problem of the encoder transmitting
information to the decoder.

2. Problem statement
Inputting an equation and a set of story keywords that learners are interested in, our goal is to generate
a Chinese math word problem in natural language. In this part, we introduce the composition of our
system and introduce the source of our data set.

Figure 1. The composition of the Chinese math word problems generation network.

2.1. System Structure
The overall framework of the CMGNN is shown in Figure 1, which consists of the following parts:

- Equation Transformer: Parse the math operation equation entered by the tutor through
  converting it into a Reverse Polish notation and removing the high-priority parenthesis
  operation. In this way, the length of the mathematical equation can be reduced and it has
certain sequential calculation characteristics for the convenience of subsequent feature
extraction.
• Keyword Collector: In order to increase the richness of the story information of the generated math word problems, the keyword collector will extract the relevant vocabulary of interest input by the learner and construct a unified keyword input format.

• Unified ED: Unified ED refers to Unified Encode-Decoder. After mixing the processed equation with the processed keywords, Unified ED will extract the mathematical relationship features and the semantic information of the input. Finally it predict the corresponding Chinese math word problem through the fusion calculation of the internal Transformer block.

2.2. Dataset creation
We create the Chinese math word problem generation dataset based on the Ape210K dataset [3]. Each sample is in the form of a triplet <E, K, T>, as shown in the Table 1.

| Table 1. An example from dataset. |
|-----------------------------------|
| Value                            |
| Equation                        | 5000+5000*3.25/100*2 |
| Keyword                         | "叔叔", "存期", "银行", |
|                                 | "年利率", "钱"]       |
| Question                        | 李叔叔把 5000 元钱存入银行，存期为两年， |
|                                 | 年利率为 3.25%，到期时，李叔叔一共能拿到多少钱? |
| Translation                     | Uncle Li deposits 5000 yuan in the bank for two years with the annual interest rate of 3.25%. |
|                                 | How much will Uncle Li get in total when it matures? |

Among them, E stands for equation, K stands for keywords, and T stands for math word problem text, which is the label of the sample. However, in the Ape210K data set, each sample does not contain keyword information. Therefore, we need to adopt a method to extract keywords from the original data. Considering that the text length of each sample is relatively short and the number of samples is large, we choose to take the dependency syntactic analysis on each math word problem, and then perform noun filtering on the processed results to obtain all nouns, which are used as keywords. Among them, we used the function interface provided by the open source library HanLP [4] for dependency syntax analysis.

3. Proposed method

3.1. Equation transformer
The equation transformer first replaces some symbols based on the rules showed in the Table 2 to standardize the input. Then, convert the infix expression into a Reverse Polish notation.

| Table 2. The rules in equation normalization. |
|-----------------------------------------------|
| Characters to be replaced | New Characters |
| X:Y                          | X/Y           |
| X%                           | X/100         |
| (X/Y)                        | X/Y           |
| X(Y/Z)                       | (X+Y/Z)       |

For example, given the following equation input:

\[ 15 +36 \times 25\% \] (1)

We replace the percent sign and generate the following equation:

\[ 15 + 36 \times 25/100 \] (2)
Then, convert it to a Reverse Polish notation to get

\[15, 36, 25, *, 100, /, +\] (3)

Particularly, the comma is an artificially inserted separator to distinguish numbers.

3.2. Keyword collector
The entered keyword K contains a set of nouns \( \{k_1, k_2, ..., k_n\} \). We put the keywords in order and insert delimiters between the words, and add a start and end flag before and after the word sequence. In this way, the cutting accuracy and semantic parsing ability of the model can be improved.

3.3. Unified ED
This part is the most important part of our system, whose core combines the methods of NEZHA [5] and UniLM [6]. Take NEZHA as the pre-training model, and design the sequence-to-sequence model architecture based on the UniLM. NEZHA obtains state-of-the-art results on most representative Chinese tasks through its relative position coding and whole word concealment strategy. Because the math word problems in this paper are also Chinese data sets, NEZHA is adopted as the pre-training model. In addition, compared with the traditional sequence-to-sequence model, the UniLM proposed by Dong does not have a fixed transfer vector so that the encoded information is transferred directly to the decoding layer through the hidden layer, which can effectively reduce the loss of encoder to decoder. In the task of math word problem generation, it involves the reasoning of deeper digital relations and logical relations, and the ability of UniLM to save information is more suitable for this task. Our method combines the advantages of NEZHA and UniLM. As shown in Figure 2, the results from Equation Transformer and Keyword Collector are stitched together with the real problem text as input to the model.

\[
\text{token} = [e, k, \text{label}] 
\] (4)

Then, a distributed representation vector \( \omega_i \) is obtained through symbol embedding layer, position coding layer and sentence coding layer.

\[
\omega_i = \text{token}_i E_\omega + P_i + S_i 
\] (5)

Under the multi-layer attention mechanism, the coding vector of the sentence hiding layer is obtained, where the attention of the input part is bi-directional, and the attention of the prediction part is unidirectional (as shown in the right half of Figure 2). The current token \( a_i \) at the time of prediction is only related to the token before the moment \( i - 1 \).

\[
q_i = W_q \omega_i 
\] (6)

\[
K = W_k \omega 
\] (7)

\[
V = W_v \omega 
\] (8)

\[
a_i = \sum_{j=1}^{i-1} (\text{softmax}(q_i K(d_k)^{-0.5})) V_j 
\] (9)

Where, \( q, K \) and \( V \) are the query value, key matrix and value matrix respectively in self-attention. After obtaining the attention weight vector, we map the vector to the probability distribution based on the vocabulary through feed forward network layer.

\[
D_i = W_2 \max \left( 0, W_d a_i + b_1 \right) + b_2 
\] (10)

Finally, find the predicted path with the highest score as the answer by the beam search algorithm.

\[
P(y|a) = \max \left( \frac{1}{n} \log \left( \prod_{k=1}^{n} p(y_k|a, y_1, y_2, ..., y_{k-1}) \right) \right) 
\] (11)
4. Experiment and discussion

4.1. Dataset
We processed APE210K to get the data set used. APE210K data set is composed of three parts, which are training set with 200,488 samples, validation set with 5000 samples and test set with 5000 samples respectively. After keyword extraction and expression normalization, the samples with the normalized solution results consistent with the original answer and the equation length less than 102 were left. The final data set used was composed as the Table 3. On average, each sample contains 4 keywords. The average equation length is 13, and the average question text length is 45.

Table 3. The count of samples from dataset.

|          | Train    | Validation | Test    |
|----------|----------|------------|---------|
| APE210K  | 200488   | 5000       | 5000    |
| Retention| 200193   | 500        | 4998    |

4.2. Evaluation method
The experiment adopts ROUGE [7] as the evaluation index, which is calculated as follows. Among them, X represents the real label, Y represents the predicted result, LCS(X, Y) represents the longest common subsequence of X and Y, and m and n are the lengths of X and Y respectively. β is a self-defined value used to adjust the weight of recall rate and precision rate. In this experiment, β is set to 8, which pays more attention to recall rate.

\[
R_l = \frac{\text{LCS}(X,Y)}{m}\tag{12}
\]

\[
P_l = \frac{\text{LCS}(X,Y)}{n}\tag{13}
\]

\[
F_l = \frac{(1+\beta^2)R_lP_l}{R_l+\beta^2P_l}\tag{14}
\]

4.3. Experimental parameters
The hyper-parameters set in the experiment are shown in Table 4, and the learning rate used by Adam to fine-tune NEZHA is 0.00002. The size of the beam search in the prediction stage is 5. The Dropout rate is 0.3 and the Batch Size of the training is 26. Experiments show that our model can converge quickly within the first three epochs.

Table 4. The hyper-parameters in the experiment.

| Hyper-parameters             | Value        |
|-----------------------------|--------------|
| Batch size                  | 26           |
| Max sequence length         | 256          |
| Max equation length         | 150          |
| Max target length           | 150          |
| Beam search size            | 5            |
| Dropout rate                | 0.3          |
| NEZHA fine-tune learning rate | 0.00002     |

4.4. Result
As a comparative experiment, we chose an end-to-end method as the baseline. The baseline model includes an encoder implemented by bi-direction GRU and a decoder implemented by LSTM. In addition, we also conducted experiments on the performance of the proposed model without the keyword collector. Finally, we selected the best model obtained in the training phase, and verified the effects of each method on the test set. The results of each model are shown in the Table 5.

It can be found from the result that the model proposed in this paper is better than the end-to-end method in the task of generating Chinese math word problems. Specifically, there is about a 12% improvement in the Rouge-L evaluation. Secondly, in terms of training time, our proposed method is
significantly superior to the end-to-end method, which can save half of the time. However, there is not much difference between the two in terms of predict time. In addition, on the basis of our model, we removed the keyword collector and performed experiments. It can be seen that the effect of the model is not only worse than the original CMGNN, but also inferior to the end-to-end model. It proves the importance of keyword fusion for the generation.

| Table 5. The rules in equation normalization |
|---------------------------------------------|
| Model          | Rouge-1 | Rouge-2 | Rouge-L | Train time | Predict time |
|----------------|---------|---------|---------|------------|--------------|
| Seq2Seq        | 0.5756  | 0.4417  | 0.5792  | 5h26m      | 2h58m       |
| CMGNN(without  | 0.4634  | 0.2652  | 0.4803  | 2h40m      | 2h28m       |
| keyword)       |         |         |         |            |              |
| CMGNN          | 0.7162  | 0.5361  | 0.7040  | 2h55m      | 2h42m       |

5. Experiment and discussion

This paper proposes a CMGNN model to generate Chinese math word problems, and shows how CMGNN generates Chinese math word problems through math equation and keywords. We strengthen the integration of logical information of math equation and semantic information of keywords. The feasibility of the encoder and decoder integration model is verified. We explore the automatic generation of Chinese math word problems, ensuring effect and low time cost. Experimental results show that the proposed model can not only achieve objective results in the quality of content generation, but also improve the training speed greatly.

In the future work, on the one hand, the discriminant correction mechanism (background knowledge, digital logic, etc.) will be introduced to further improve the effect and customization of the model. On the other hand, we will explore the computational optimization of the model to reduce complexity and time cost.

References
[1] Bahdanau D, Cho K and Bengio Y 2014 Neural machine translation by jointly learning to align and translate Computer Science
[2] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, ... and Polosukhin I 2017 Attention is all you need Preprint arXiv:1706.03762
[3] Zhao W, Shang M, Liu Y, Wang L and Liu J 2020 Ape210K: A large-scale and template-rich dataset of math word problems Preprint arXiv:2009.11506
[4] He H 2014 HanLP: Han language processing URL: https://github.com/hankcs/HanLP
[5] Wei J, Ren X, Li X, Huang W, Liao Y, Wang Y, ... and Liu Q 2019 NEZHA: Neural contextualized representation for Chinese language understanding Preprint arXiv:1909.00204
[6] Dong L, Yang N, Wang W, Wei F, Liu X, Wang Y, ... and Hon H W 2019 Unified language model pre-training for natural language understanding and generation Preprint arXiv:1905.03197
[7] Lin C Y 2004 Rouge: A package for automatic evaluation of summaries Text summarization branches out pp 74-81