Recurrent Convolution Attention Model (RCAM) for Text Generation based on Title

YUAN JIANGLIN¹, GUO ZHIGANG¹, a and CHEN GANG¹

¹The Information System Engineering Institute, Information Engineering University, 450000 Zheng Zhou He Nan, China

a Corresponding author: keep_ep_er@163.com

Abstract. Natural Language Generation (NLG) is one of the most important part in Natural Language Processing (NLP). Recently, generating text automatically with deep learning method has been improved a lot. While there are lots of defects in text generation such as the quality is not satisfied and the text of title is not clear. The paper used the recurrent convolution attention model with LSTM (Long Short-Term Memory) cells for text generation by giving a title. The result proved that it can generate sentence according with the title and make the text express more fluently. Moreover, it uses less time to train by contrast with the SeqGAN (Sequence Generative Adversarial Networks). At the same time, the result is better than other attention mechanism with LSTM models. Therefore, it has more significance for NLP research.

1. Introduction

Both text-to-text generation and data-to-text generation are instances of Natural Language Generation (NLG) [1]. In this paper, we research more on text to text, so called text generation. Natural Language Processing is difficult for research because the characters are discrete data and the expression of language is multiple. However, deep learning algorithm is better on the fitting of high-dimensional functions than the traditional methods. It is a good representation on complicated function and the better classification on multiclass problem that makes deep learning used widely in natural language processing.

There are a lot of types basic cells such as RNN and LSTM, CNN. RNN and LSTM can learn the relationship between two characters well. However, they are not good at learning the relationship between sentence with lots of words. While there are a lot of models created by these basic cells such as Encoder-Decoder model. Encoder-Decoder model contains an encoder and a decoder. RNN is always used to be as encoder and decoder in NLP. It encodes the input into a vector representation. The vector serves as the auxiliary input to a decoder RNN [1]. Sequence to Sequence is another Encoder-Decoder model which is mainly used in machine translation. It also can be used in text generation. While the text generated by Encoder-Decoder framework is not flexible. Generative Adversarial Networks (GAN) is another deep learning models which consists of a generate model and a discriminate model. It is mainly used in continues data training problem. Therefore, it was not used widely in NLG because of the discrete data in NLP, until researchers improved the training policy such as SeqGAN [2], LeakGAN [3], FairGAN [4]. However, GAN is not stable enough to train in NLP. Moreover, it does not use prior information which makes it cost a lot to train good results. Attention mechanism is another method to solve the problem that history information vanished in RNN or LSTM model [5]. It can make good use of history information which is used to generate the
Convolutional Neural Networks (CNN) is a good architecture to extract features in image processing. It can also be used in NLP [6]. Convolutional neural network (CNN) and recurrent neural network (RNN) are two main types of DNN architectures, which are widely explored to handle various NLP tasks. CNN is supposed to be good at extracting position-invariant features and RNN at modeling units in sequence [7].

This paper combined the advantages of CNN and LSTM models with attention mechanism and embedded topic to generate text recurrently. On the one hand, it can make good use of history information and also learn the sentence structure by latent variables. On the other hand, it will use less resource to train and the model is more stable. At the same time, it makes the generated text more flexible with the specific topic.

2. Modelling Approach

To begin with, we state the text generation problem as follows. We define input $x_t = \{y_0, ..., y_{t-1}, T, \alpha\}$ at time $t$. Variable $T$ represents the title variable and variable $\alpha$ means the structure latent variable. The goal of text generation is to generate the sequence $Y_{1:m} = (y_1, ..., y_t, ..., y_m)$. Variable $m$ means the length of sequence. The word in sequence is generated by formulation (1).

$$p(t) = \max \{p(y_t | y_1, ..., y_{t-1}, T, \alpha)\} \quad (1)$$

We will use the maximum likelihood probability to generate the next word at time $t$.

2.1. LSTM Model

An LSTM network contains LSTM units in RNN and an LSTM unit is a recurrent network unit that excels at remembering values for either long or short durations of time [8-9]. It contains an input gate, a forget gate, an output gate and a memory cell [10]. Respectively, at time $t$, we set the above parts as $i_t$, $f_t$, $o_t$, $c_t$. In an LSTM network, we propagate as Equation (2)(3)(4).

$$\begin{align*}
\begin{bmatrix} i_t \\ f_t \\ o_t \end{bmatrix} &= \text{activation}\left( \begin{bmatrix} W_i \\ W_f \\ W_o \end{bmatrix} \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} \right) \quad (2) \\
\begin{bmatrix} c_t \end{bmatrix} &= i_t \cdot \text{activation}\left( W_c \cdot \begin{bmatrix} h_{t-1} \end{bmatrix} \right) + f_t \cdot c_{t-1} \quad (3) \\
h_t &= c_t \cdot o_t \quad (4)
\end{align*}$$

2.2. Recurrent Convolution Attention Model

Recurrent Convolution Attention Model (RCAM) can make good use of history information. The basic model is shown in Figure 1.

![Figure 1. The structure of RCAM.](image)

As shown in Figure 1, title and latent variables are embedded as input for LSTM cells. After LSTM generated word at time t, the generated word from time 0 to t will be convoluted recurrently to get history information and the structure of sentence. Then the convolution result will be used to
calculated the attention result with LSTM output at time \( t \). It will be as the input for next LSTM at time \( t+1 \). The initial input for LSTM at time 0 can be formulated with Equation (5).

\[
y_0 = f(T, \alpha)
\]

(5)

The output of LSTM at time \( t \) is formulated with Equation (6).

\[
y_t = \text{LSTM}(\text{attention}(y_{t-1}, c))
\]

(6)

Variable \( c \) in Equation (6) represents the result of recurrent convolution with the output words at time 0 to \( t-1 \). While the recurrent convolution algorithm is shown in Table 1.

As shown in Table 1, recurrent convolution algorithm can update the information of generated text. It can get over the disappearance of history information weakness in NLG by combining with the attention mechanism. The attention mechanism is formulated as Equation (7).

\[
\text{attention}(y_{t-1}, c) = \tanh(y_{t-1}U + cV)
\]

(7)

### Table 1. Recurrent Convolution Algorithm

| Algorithm 1: Recurrent Convolution Algorithm |
|---------------------------------------------|
| 1: Input the generated text \( S = \{y_0, ..., y_{t-1}\} \) |
| 2: for \( t - 1 \) do |
| 3: \( c = \text{conv}_{-1}d(S) \) |
| 4: end for |
| 5: return \( c \) |

3. Experiments

3.1. Dataset

Text generation task needs the high-quality corpus. At first, we collected a lot of text with high-quality from Internet. They were mainly from essays which got high scores. Then we preprocessed the corpus by selecting sentence which fitted the requirement length. At the same time, we used TextRank [11] method to extract the title of sentence. However, it had noise with this way. Therefore, we collected text from ZHIHU website and preprocessed with the same way.

3.2. Settings

We used tensorflow packages in python language programming. Some hyperparameters were set as follows. We selected almost 61852 high-frequency words for the vocabulary on training and testing. We also used the word2vec [12] to train the word-vectors. The dimension of word-vectors was set to 100. The size of hidden layers in LSTM cell were set to 500. Model hyperparameters were initialized to uniform distribution from -0.05 to 0.05. We set the dropout as 0.5 in neural networks and used the AdaDelta algorithm to update the parameters in neural networks.

3.3. Results

We compared different models by training the corpus. We gave five keywords as titles: spring, hope, culture, China, happy. The results were shown in Table 2.

As shown in Table 2, three models can generate the text. However, SeqGAN generated text which were not fitted to the topic mostly. Attention LSTM model generated text with some syntax error. Moreover, some text could not fit to the topics. While the RCAM could generate the text with
Table 2. SeqGAN, Attention LSTM and RCAM generated text with different topics.

| Title  | Model  | Generating Sentence                                                                 |
|--------|--------|--------------------------------------------------------------------------------------|
| Spring | SeqGAN | Fish is playing <unk> with bubbles happily in the water … <pad>                      |
|        | AttLSTM| Spring is coming <unk> and snow are melting, which means everything is recovering … <pad> |
|        | RCAM   | The spring is beautiful with warm sunshine and blooming flowers.                     |
| hope   | SeqGAN | Youth is the hope for the future of our country . . . <pad> <pad> <pad> <pad>       |
|        | AttLSTM| I hope that seniors give me a hand.                                                 |
|        | RCAM   | People sow seeds in spring and hope for the harvest in autumn.                      |
| culture| SeqGAN | I love the beautiful country and it brings us happiness. . .                          |
|        | AttLSTM| The motherland <unk> has thousands of years history and culture, <pad> which is worthy of everyone's admiration. |
|        | RCAM   | China is beautiful and has long history, <pad> <pad>                                 |
| China  | SeqGAN | Longing for peace is a yearning and a pursuit. <pad>                                |
|        | AttLSTM| China has realized an Olympic dream, which is a special dream. <pad>                |
|        | RCAM   | China has a peace dream which is green.                                              |
| happy  | SeqGAN | Feeling happy is real which likes the warm sunshine. <pad> <pad>                    |
|        | AttLSTM| Happiness is a healthy attitude <unk> which everyone should pursue in the world. <pad> |
|        | RCAM   | Reading more can also make you happy. <pad>                                          |

Specific topics and less syntax error than Attention LSTM. At the same time, it used less time to train compared with SeqGAN model. The SeqGAN model was trained 1,000,000 times to get the result.

We also used BLEU metrics and ROUGE metrics to evaluate the generated text. BLEU (bilingual evaluation understudy) [13] is an algorithm for evaluating the quality of text which has been machine- translated from one natural language to another. It contains BLEU-2, BLEU-3, BLEU-4 etc. BLEU was one of the first metrics to claim a high correlation with human judgements of quality [14-15], and remains one of the most popular automated and inexpensive metrics. BLEU’s output is always a number between 0 and 1. This value indicates how similar the candidate text is to the reference texts,
with values closer to 1 representing more similar texts. We used BLEU-2, BLEU-3 and BLEU-4 for evaluate the generated text.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [16] is a set of metrics and a software package used for evaluating automatic summarization and machine translation software in natural language processing. The metrics compare an automatically produced summary or translation against a reference or a set of references (human-produced) summary or translation. It contains ROUGE-N, ROUGE-L, ROUGE-W and ROUGE-S. We used ROUGE-L as the evaluation metric for generated text. Results were shown in Table 3.

As shown in Table 3, RCAM had better score in BLEU metrics and ROUGE metrics evaluation. On the one hand, RCAM used the recurrent convolution to get history information and the text structure information. On the other hand, LSTM cells can learn the relationship between two words in RCAM. With combined advantages of two neural networks, RCAM generated better text than other two methods.

| Model   | BLEU-2 | BLEU-3 | BLEU-4 | ROUGE   |
|---------|--------|--------|--------|---------|
| SeqGAN  | 0.43   | 0.43   | 0.37   | 0.48    |
| AttLSTM | 0.41   | 0.40   | 0.36   | 0.44    |
| RCAM    | 0.45   | 0.41   | 0.38   | 0.49    |

4. Conclusions

This paper mainly researched the text generation by deep learning algorithms. We explained the problems of Natural Language Generation with different models by using deep learning algorithms. We analysed some models in deep learning algorithms in Natural Language Processing.

And then we proposed a new method called RCAM (Recurrent Convolution Attention Model). Compared with SeqGAN and Attention LSTM, RCAM had better result because it combined the advantages of CNN and LSTM. We then created our own corpus by collecting corpus from Internet. Before we used corpus for training and testing models, we pre-processed our corpus by clean text and extracted titles for training data. The results showed that RCAM was better than the SeqGAN and Attention Model in BLEU and ROUGE scores. The results can fit the theoretical analysis.

All in all, it will be significant for NLP research in the future. Although, it was also weakness for long term dependencies and also has some syntax error in generated text. These drawbacks will be researched in future.

Reference

[1] Gatt A, Krahmer E. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation[J]. Vestnik Oftalmologii, 45 (2018).
[2] Yu L, Zhang W, Wang J, et al. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient[J]. (2016).
[3] Guo J, Lu S, Cai H, et al. Long Text Generation via Adversarial Training with Leaked Information[J]. (2017).
[4] Xu D, Yuan S, Zhang L, et al. FairGAN: Fairness-aware Generative Adversarial Networks[J]. (2018).
[5] Parikh A P, Täckström O, Das D, et al. A Decomposable Attention Model for Natural Language Inference[J]. 2249-2255, (2016).
[6] Bai S, Kolter J Z, Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling[J]. (2018).
[7] Yin, Wenpeng, Kann, Katharina, Yu, Mo, and Schutze, Hinrich. Comparative study of CNN and RNN for natural language processing. arXiv:1702.01923, (2017).

[8] Graves A. Supervised Sequence Labelling[M]// Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg, 5-13 (2012).

[9] Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling[C]// Interspeech. (2012).

[10] Dong L, Huang S, Wei F, et al. Learning to Generate Product Reviews from Attributes[C]// Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. (2017).

[11] Peng L, Wang B, Shi Z, et al. Tag-TextRank: A Webpage Keyword Extraction Method Based on Tags[J]. Journal of Computer Research & Development, 11 (2012).

[12] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. 26 (2013).

[13] Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]// Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 311-318(2002).

[14] Papineni K, Roukos S, Ward T, et al. Corpus-based comprehensive and diagnostic MT evaluation: initial Arabic, Chinese, French, and Spanish results[C]// International Conference on Human Language Technology Research. Morgan Kaufmann Publishers Inc. (2002).

[15] Callison-Burch C. Re-evaluating the Role of Bleu in Machine Translation Research[C]// Conference of the European Chapter of the Association for Computational Linguistics. (2006).

[16] Lin C Y. ROUGE: Recall-oriented understudy for gisting evaluation[J]. (2003).