International Core Journal of Engineering Volume 7 Issue 5, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

Research on Text Generation Based on LSTM

Lifen Li[1,a], Tianyu Zhang[1,b]

[1]School of Control and Computer Engineering, North China Electric Power University, Baoding 071000, China.

[a]lilifen70@163.com, [b]1660384423@.com

Abstract

Text generation is an important direction in the field of natural language processing (NLP). In the era of pre-training model, transformer improved pre-training text generation model still can not achieve relatively ideal results, and at the same time, there is no efficient language model to automatically evaluate the quality of generated text. As an improved variant of cyclic neural network (RNN), long-term memory network (LSTM) is characterized by its long-term dependence, and it performs very well in the task of processing long sequences. The LSTM has many improved variants for different tasks, including the gated loop unit (GRU) and the LSTM with a peephole connection, all of which have better performance than the LSTM in specific tasks. However, it is not clear whether these improved variants have better performance in the field of text generation. Therefore, an exploratory text generation experiment is conducted to solve this problem. By comparing the generated text quality of standard LSTM with LSTM's improved variant GRU and LSTM model with visual hole connection, the evaluation results of LSTM model in long text field are obviously better than those of the other two models through three evaluation indexes: confusion degree, BERT score and BLEURT. Finally, we draw a conclusion and research direction that the native LSTM in the field of long text still has very superior performance. In the future, we can design a pre-training model based on LSTM for text generation. Future language models can be designed to guide the optimization and improvement of language models through large-scale evaluation using automated evaluation indicators such as BERT score and BLEURT, which are close to manual evaluation, so as to design language models that can generate higher quality text.

Keywords

LSTM; Peephole Connection; GRU; BERT Score; BLUERT.

1. Introduction

Text generation is an important aspect of natural language processing, and it is the most difficult and difficult scientific problem to explain. In recent years, the application and demand of text generation has been increasing. Tencent's Dream writer script-writing robot and today's headline-making xiaoming bot news-writing robot and other text-generation robots are constantly emerging. For a moment, they have become one of the hottest topics. The principle of machine news writing is to use the neural network model to generate new news stories based on old news texts and data [1].

Automatic text generation can include text-to-text generation (text-to- text generation), meaning-to-text generation (text generation), data-to-text generation (data-to- text generation) and image-to-text generation (image-to- text generation)[2].Among them, the principle of text-to-text generation technology is to learn the existing old text to generate new text, such as to transform the characters and plots of the old story set into a more vivid and interesting new story, to realize the processing and utilization of text. This text generation experiment uses text-to-text generation.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

Although the field of text generation is developing rapidly and various model structures emerge endlessly, the quality of generated text is still difficult to reach the level of human writing and there is no uniform evaluation standard. At present, the model with the highest quality of text generation is the pre-training model GPT-3 with transformer decoding structure. Although the GPT series model has achieved good results, when putting the text generated by GPT and the normal text written manually together, one can easily determine which author the text belongs to, which indicates that the current language model can not pass the Turing test [3].But for the generation of prose, novels and other long texts containing human emotions, the current research is still in the primary stage. Deep and high-parameter GPT model fails to solve these problems, which also makes the development of text generation almost stagnate. There are two main reasons: one is that GPT based on transformer is inefficient to deal with the strict sequence of tasks before and after the text, and there is structural difference between the base model and the specific field. The second is the lack of effective evaluation indicators to guide the optimization and improvement of the model. In many cases, researchers are unable to determine whether their improvement measures really lead to the improvement of the model performance. When the number of texts is large, manual evaluation is faced with a series of problems caused by excessive time cost. In fact, the difficulty of evaluating language models has always been the key factor restricting the development of natural language generation (NLG), and the accuracy and flexibility of manual evaluation has always been the most convincing evaluation index, but manual evaluation has a great defect of slow evaluation speed. As the most commonly used evaluation index of language model, the degree of confusion is often used as an important reference because the evaluation is relatively objective and fast, but sometimes the text with low degree of confusion but poor quality is evaluated manually. When the number of text is large, manual evaluation will consume a lot of time, which makes the model training have to stop halfway, which directly leads to the development of language model subject to the slow speed of manual evaluation. In order to make the automatic evaluation result of the machine closer to the manual evaluation result, there are new evaluation indexes based on the pre-training model BERT, such as ERT score, BLEURT, and the accuracy of these new evaluation indexes has been proved in many experiments.

The earliest improvements in text generation are actually the use of the improved version of LSTM of RNN, which has unique three-door and two-state structural features that make it stand out in the NLP field. In the development history of LSTM, there are many improved versions for LSTM, such as GRU, LSTM with peephole, which perform better in some specific areas. In addition, the poor parallelism of LSTM makes it cold in the training model era. As technology continues to evolve and new changes have emerged in many studies, we need to revisit the role of LSTM.

This study creatively evaluated the quality of text generated by different models using new evaluation metrics, and aimed to find the best base model by comparing the performance of LSTM with its variant GRU and LSTM with visual hole connections in the long text field. In order to ensure that the generated text can be compared at a high quality level and to maximize the subtle performance differences between different models, CBOW is also used in all models to improve the quality of word vectors, and cluster search is used to aid prediction.

2. Pair of LSTM neural network model improvement and optimization

2.1 Model structure of original LSTM

A major feature of the long-term memory network (LSTM) model is that it can store long-term dependencies and allow the information of the previous time to participate in the current calculation [4]. In order to solve the problem of long-term memory decline caused by gradient disappearance, LSTM creatively uses the three structures of input gate, forgetting gate and output gate, in which the input gate controls the degree of the current input read into the cell, forgetting gate determines how much of the previous cell state will enter the current cell state, and output gate converts a certain amount of cell state into hidden state. The gating mechanism allows the gate to be selectively input and output by activating the function outputs between 0 and 1. Where 0 indicates that the cell is

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

inhibited, any information can not be passed through this gate,1 indicates that the cell is activated, data can be passed directly, but most of the time the activation function outputs a value between 0 and 1, which can be understood as a weighting factor. Gating mechanism of LSTM can filter out some previously useless information, retain the most useful memory for the model, effectively solve the problem of gradient disappearance, make LSTM break through the performance bottleneck of traditional circular neural network (RNN) to store long-term memory, and realize the storage of long-term memory. The internal logic diagram of the LSTM is as follows:

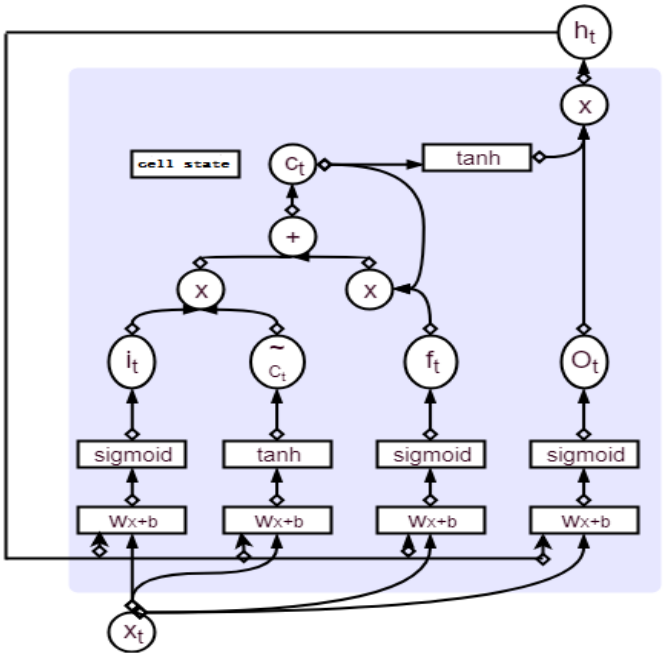


Figure 1. Cell structure of LSTM

$$\text{Input gate: } i_t = \sigma\left(W_{(ix)}x_t + W_{(ih)}h_{(t-1)} + b_i\right) \quad (1)$$

$$\text{Candidate vector: } \tilde{c_t} = tanh\left(W_{(cx)}x_t + W_{(ch)}h_{(t-1)} + b_c\right) \quad (2)$$

$$\text{Forget gate: } f_t = \sigma\left(W_{(cx)}x_t + W_{(ch)}h_{(t-1)} + b_f\right) \quad (3)$$

$$\text{Current cell state: } c_t = f_t c_{(t-1)} + i_t\tilde{c_t} \quad (4)$$

$$\text{Output gate: } o_t = \sigma\left(W_{(ox)}x_t + W_{(oh)}h_{(t-1)} + b_o\right) \quad (5)$$

$$\text{Final cell state: } h_t = o_t\, tanh(c_t) \quad (6)$$

According to the figure, the formula can be understood intuitively. In the formula, s represents the activation function sigmoid, while tanh corresponds to the candidate vector, W and b respectively represent the weight and bias. Detailed calculation process is:

1) First, the input gate, candidate vector, forget gate and output gate are calculated respectively with the current input x_t and the last hidden state $h_{(t-1)}$ of the previous period.

2) The current cell state is then calculated using the input gate, the forgetting gate, the candidate vector, and the cell state of the previous time.

3) Finally, the final cell state (hidden state) is calculated from the output gate results and the current cell state.

Output of LSTM model:

The resulting cell state is not the end result we want, we need to use individual cell states for output or prediction. As a result, we use the softmax layer at the top level to calculate and output.

$$y_t = soft\,max(W_s h_t + b_s) \quad (7)$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

The final image of the LSTM is as follows:

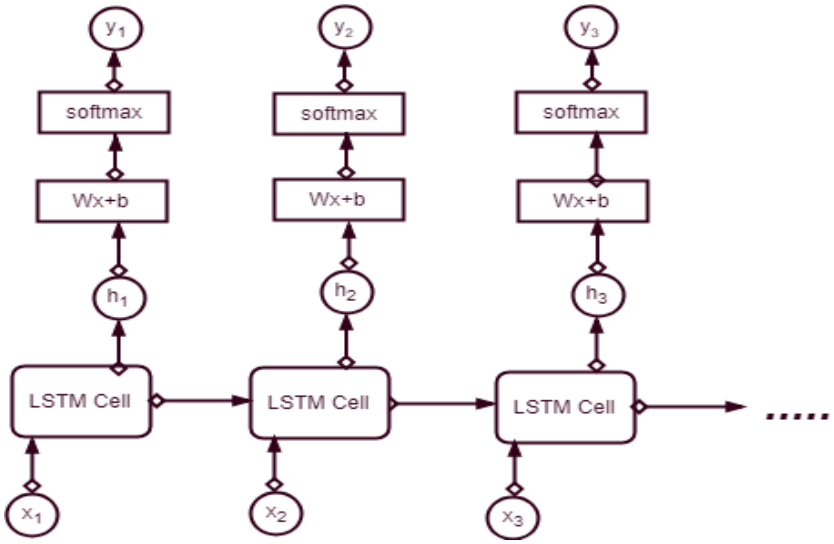


Figure 2. Complete structure diagram

2.2 LSTM with peephole connection (Peephole Connection)

Gers and Schmidhuber put forward the idea of adding peephole to LSTM, that is, let the input gate and forget gate see the cell state of the previous moment [5].In this experiment, a diagonal peephole connection is used, which can better adapt to the language model as an improved variant [6].Fixed the defect that the input gate and the forgetting gate in LSTM can only see the current input and the final hidden state, and discarded some information in the cell at the last moment. But it also increases the number of parameters that the model needs to calculate.

The structure of the model is as follows: Adds the current cell state to the output gate. As a result, peephole connections provide more control over cell state. It has been proved that peephole connection has achieved very good results in many applications and has been widely used in Chinese word segmentation model [7]. Dark black in figure represents peephole connection.

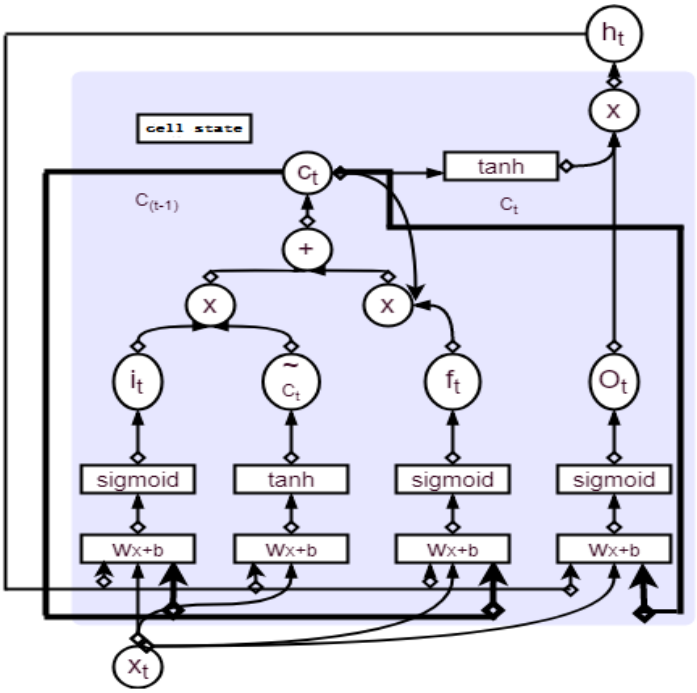


Figure 3. Structure diagram of LSTM with peephole connection

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

2.3 Gated Recurrent Unit (GRU)

GRU makes a large adjustment on the structure of LSTM, which is a very elegant variation, simplifying the structure without damaging the performance of the model [8]. The LSTM model has three gates and two states, which means that the training model requires a large number of parameters and the GRU reduces the number of parameters of the LSTM.

GRU unit structure (above) and LSTM unit structure (below) are shown in the figure, which can visually show the difference between the two cells.

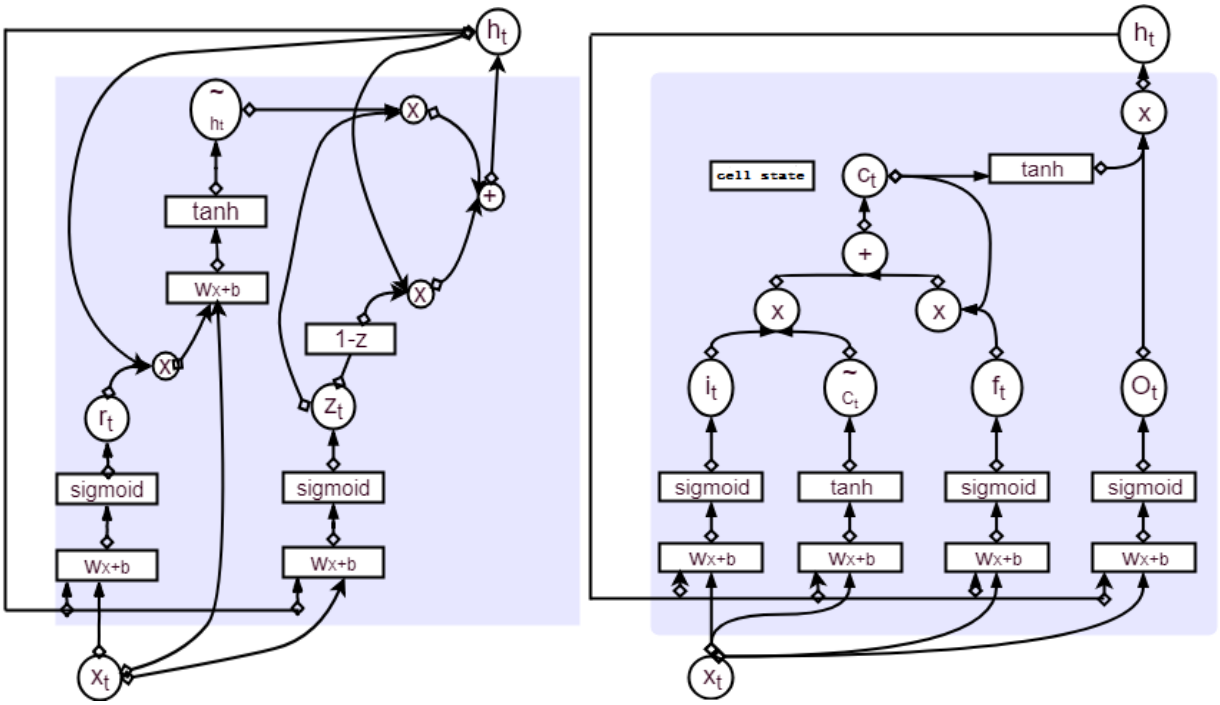


Figure 4. Structure Comparison between GRU and LSTM

GRU changes the input gate, forget gate and output gate in LSTM into two gates to control the flow of information, namely reset gate and update gate. Among them, the calculation formula of reset gate is as follows:

$$r_t = \sigma\left(W_{(rx)}x_t + W_{(rh)}h_{(t-1)} + b_r\right) \tag{8}$$

$$\tilde{h}_t = tanh\left(W_{(hx)}x_t + W_{(hh)}\left(r_t h_{(t-1)}\right) + b_h\right) \tag{9}$$

If the value of r_t is 0, the value of $r_t h_{(t-1)}$ is also 0, and the previous hidden state $h_{(t-1)}$ is ignored during calculation. When the reset gate is close to 1, the previous complete information will be read in. The final status is calculated by the update as follows:

$$z_t = \sigma\left(W_{(zx)}x_t + W_{(zh)}h_{(t-1)} + b_z\right) \tag{10}$$

$$h_t = z_t\tilde{h}_t + (1 - z_t)h_{(t-1)} \tag{11}$$

In standard LSTM, the input gate determines how much of the current input is read into the cell state, and the forgetting gate determines how much of the previous state is read into the current state. In GRU, the input and forget gates are combined into an update gate. If the update gate z_t is 0, the current input through the reset gate will not enter the current state, and the current state h_t is still the state $h_{(t-1)}$ of the previous moment. The opposite is true when the update gate is 1.

Graph structure and calculation show that GRU has only one final state, while in LSTM, there are two states: cell state and final hidden state. As a result, the output gate does not use space, and the GRU without the output gate reduces the number of parameters to be calculated.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

2.4 Cluster Search

The role of cluster search in text generation is to help the model generate text better during the test phase and not during the training phase. In the process of exporting words, if the most likely word is output every time, the final output of the model must be very single, which makes semantic fluency and grammatical integrity very bad. Cluster search comprehensively evaluates the words that may appear in the future and the current words, obtains the words with the highest semantic similarity, and improves the prediction quality of words [9].

Core idea: The structure of cluster search can be regarded as a tree. It is necessary to find k paths from root node to leaf node and satisfy the maximum joint probability value of node. The length of path can be regarded as the length of cluster.

Process: First, select a word as the starting node, and each time step outputs a fixed sequence. And then, the joint probability is calculated according to the sequence of the current time step and the sequence of the previous time step, and k sequences with the largest probability value are taken as the best candidate, and then the number of times that the best candidate is used for repeated searching until the given cluster length. Finally, output k of the best phrases, which are the clusters obtained by searching.

2.5 Word embedding method Word2Vec

One-Hot encoding (single-hot encoding) sets words in text that belong to a category to 1 and other words to 0, generating a word vector. One-Hot encoding is simple, intuitive and easy to implement, and performs well when dealing with features that are less relevant.

In the field of text generation, characteristic engineering needs to learn the inherent logical relationship between different words in text data set. If you use One-Hot encoding, because the dot product of different words is always 0 when calculating the dot product between vectors, you can not actually express the relationship between word vectors in this way. In addition, the use of One-hot encoding model parameters increases the risk of overfitting.

Mikolov et al. proposed the Word2Vec model, which can dig the semantic relationship between word vectors to a greater extent, and obtain the low-dimensional feature representation of words [10]. Word 2Vec is a simple neural network that includes Skip-Gram and CBOW models. Where Skip-Gram gives the target word to predict the context word and CBOW is the context word to predict the target word. The specific training process is as follows:

1) Find the word vector corresponding to the input word from the word embedding layer.

2) Input the word vector into the neural network to get the correct prediction output.

3) Calculate the loss of forecast and real words.

4) Optimize neural networks and word embedding layers based on loss functions and optimizers.

5) Generate a valid probability distribution using the softmax activation function.

6) In this experiment, the CBOW model is used to learn the word vector because the CBOW algorithm performs better on small data sets than Skip-Gram [11].

3. Text Generation Process

Before using LSTM, use CBOW in word2vec to get high-level feature vectors, which can greatly improve the quality of text generation. Cluster search also improves the quality of text generation during the prediction phase.

Algorithm flow of the model is as follows:

1) Break the text into words and store it in a list of multiple documents, each of which is a list of words.

2) Convert the list containing multiple documents to digital vector, input CBOW model to obtain low-dimensional dense vector and save it as.npy file, and import it directly from the file when data is needed.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

3) Define batch data generation in the form of One-hot encoding.

4) Replace the dimension corresponding to the word vector encoded by One-hot with the dimension converted high-level purification feature of CBOW output.

5) Define the parameters of LSTM or other variants and input the characteristics of CBOW output into LSTM.

6) Calculate the cross entropy loss of the model.

7) Gradient cropping prevents the gradient from exploding by defining the learning rate and optimizer by gradient cropping.

8) Run the LSTM using the cluster search method to generate text.

Here's the general flow of text generation.

4. Experiments

4.1 Experimental data

In order to eliminate the randomness of the experiment,100 consecutive texts were randomly selected.

Processing data:

1) Read the 100 texts, add each text to a file list in the form of a list, and get a list of 100 documents with different lengths.

2) Traverse the list containing multiple documents to turn it into a long list, count the number of times each word appears, delete the words less than 10 times, and obtain a key value pair with the word as the key and the number as the value.

3) Repeat the list obtained in 1) again and replace each word in the list with the corresponding number. If the number of words in 2) is too small to be deleted, replace it with "UNK ".

4) Reverse the key value pair to obtain a number of times as the key, and the word is a new key value pair as the value. Finally, the output value of the model can find the corresponding number through the lower value of the maximum probability value, and then the number can take out the corresponding word from the dictionary.

Input and output of CBOW

Input: the list of digital vectors including 100 documents, data_list, word vectors of key value pairs, and the length of each word vector is 128.

Output: Size corresponds to input, output is dense vector with high semantic similarity.

Dataset:

Training set: one-hot encoding vector after batch processing of data_list in bigram and high-level vector predicted by CBOW in word2vec-based model.

Validation Set: The first 10 documents with more than 1000 words in data_list are used as validation data and are also batch processed. Similarly, if CBOW is used, the output vector of CBOW is used as the verification set.

4.2 Experimental parameters

In order to enhance the training effect of the model and improve the quality of text generation, the superparameters needed for this experiment are defined. The LSTM and its variant superparameters are consistent, so that the experimental results can be directly compared. The specific parameters are as follows:

Table 1. LSTM hyper-parameters table

Properties	Parameters
Number of neurons	128
Amount of data processed per step	64
Steps	50
Regularization coefficient *(dropout)*	0.2

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

Table 2. CBOW parameter table

Properties	Parameters
Sample size per batch	128
Dimension of embedded vector	128
Context Window Size	3

Table 3. Parameter table of cluster search

Properties	Parameters
Cluster length	5
Number of candidates	5

In order to achieve better performance and reduce overfitting, the experiment adopts the attenuation learning rate. Different from the fixed learning rate parameter, the attenuation learning rate changes with the continuous development of model training, and when a certain condition is reached, the learning rate decreases. In this experiment, we set the learning rate to be reduced to 0.5 times of the original without reducing the degree of confusion.

4.3 Model Evaluation Indicators

Whether the model can be accurately evaluated and whether the direct relationship conclusion is correct or not, the scientificity and effectiveness of the evaluation index become an important guarantee for scientific and effective experimental research. The inefficient evaluation index will lead to wrong research direction and even draw completely wrong conclusions. Human evaluation is still the most accurate evaluation index in NLP processing tasks such as text generation and machine translation, and can be regarded as the last criterion of all evaluation indexes. However, when the data set is large, the model is complex, and the generating task is large, the manual evaluation will consume a lot of manpower and material resources, and even make the model training have to stop, which is unacceptable for the complex model training that takes months. As a result, the automatic machine evaluation index close to manual evaluation becomes the key factor for the rapid development of NLG field.

At present, the innovation and improvement of evaluation indexes are based on the principle that the closer the machine evaluation is to the manual evaluation, the better the evaluation indexes are. However, the overlapping evaluation indexes BLUE and ROUGE based on N-gram are based on the changes of words and phrases. Compared with the manual evaluation which takes into account the criteria of grammar accuracy, statement flow smoothness and semantic similarity, there is still a big gap. [14-16] In response to the problem of difficult evaluation of language model, relevant researchers have done a lot of exploratory research, and the recent success is based on the scoring index of pre-training model Bert, such as BERT score [12], BLEURT proposed by Google researchers [13].

1) BERT score (Evaluating text generation with Bidirectional Encoder Representations from Transformers)

Basic idea: use Bert to extract features for a set of corresponding reference sentences and candidate sentences, and then calculate the inner product of each word in the reference sentence and each word in the candidate sentence to obtain the similarity matrix. By using this matrix, the maximum similarity score of this set of sentences is accumulated and normalized to obtain the accuracy rate, recall rate and F1 value of BERT score.

2) BLUEURT (Bilingual Evaluation Understudy with Representations from Transformers)

BLUEURT is a new evaluation method for text generation. It has achieved very excellent performance through two pre-training and two fine-tuning to fit manual evaluation with higher precision. In the 2019 WMT indicator sharing task, BLUEURT is nearly 48% more accurate than BLUEU [13]. BLUEURT no longer simply computes the similarity between words, but pays more attention to semantic similarity and achieves high-precision evaluation.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

Basic idea: preheat the target of the language model, i.e. the first pre-training, then perform the second pre-training on the synthetic evaluation target, then perform the first fine adjustment on the WMT indicator sharing task, and finally perform the second fine adjustment for the specific data set to realize the complete end-to-end training process.

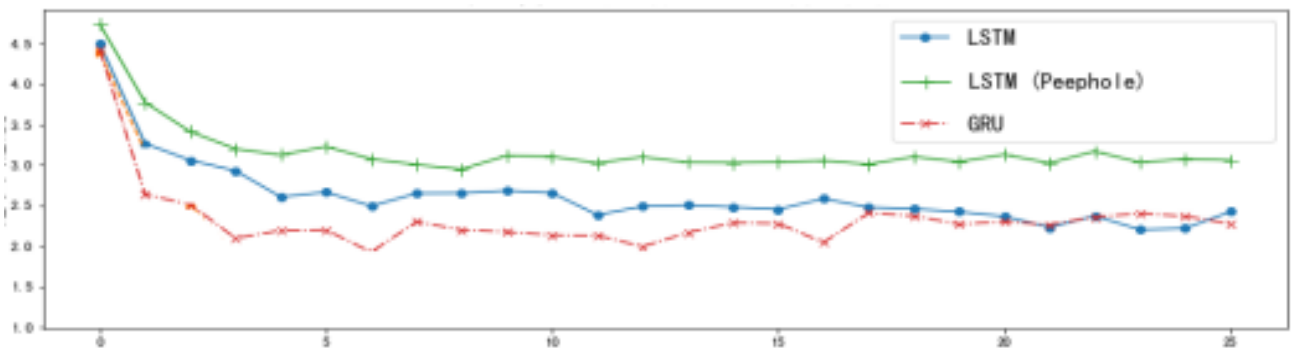
3) Perplexity

Confusion degree is an important index to measure the advantages and disadvantages of language model. Its basic idea is that the language model with higher probability value is better for the sentence of test set. When the language model is trained, the sentence in test set is normal, then the higher the probability of the trained model on test set is better. In actual performance, the lower the degree of confusion, the better the model.

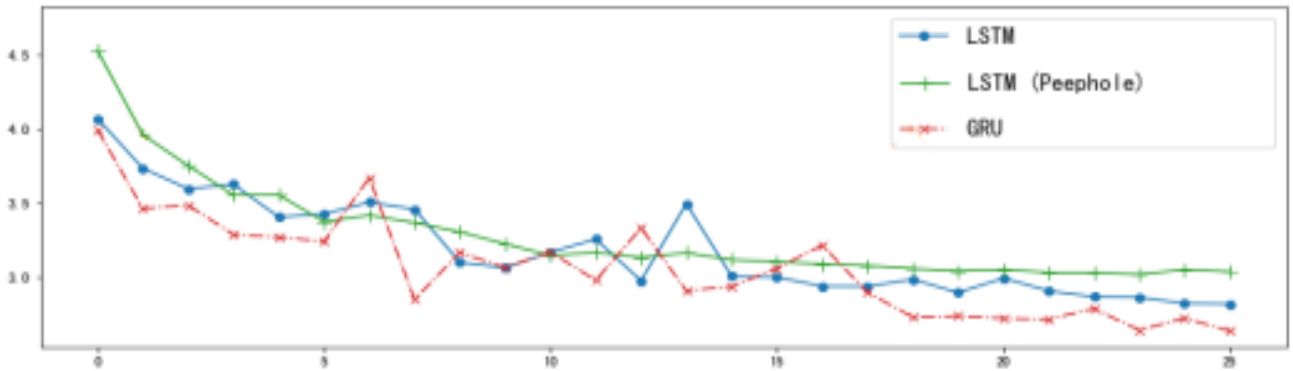
4.4 Experiment implementation

In order to compare the performance of different cyclic neural networks in the field of long text generation, it is necessary to keep the consistency of other variables except cyclic neural networks in the experiment. Therefore, CBOW in word2vec is used as a method of word embedding and cluster search is used to help predict. Three kinds of cyclic neural networks, namely standard LSTM, GRU and LSTM with visual hole connection, are trained respectively, and the contrast of text quality generated in the training process is presented in an intuitive way.

4.5 Experimental results and analysis



(a) The perplexity performance of the training data of various variants of LSTM



(b) The perplexity performance of the Validation data of various variants of LSTM

Figure 5. Variation of Perplexity of Different Models with Time

Table 4. Comparison of BERT score and BLEURT evaluation indicators of different models unit:%

Model	BERT score Score			BLEURT Score
	P	R	F1	
LSTM (peephole)	68.25	67.54	67.89	52.24
GRU	78.45	78.13	78.29	71.34
LSTM	82.34	83.21	82.77	77.63

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

Experimental analysis: LSTM with peephole connection can use the previous cell state through the peephole, and theoretically can obtain more information of the previous period. GRU combines the input gate and the forgetting gate into a new gate, and combines the cell state and the final hidden state of LSTM into a single hidden state, minimizing the number of parameters, reducing the possibility of overfitting, making the convergence speed of the model faster.

It can be seen from the figure that the confusion degree of LSTM with peephole is the highest in both the training set and the verification set, the lowest in the training set and the lowest in the verification set.

In the accuracy rate Precision, recall rate Recall and F1 values of BERT score, LSTM scores were significantly better than GRU and LSTM (peephole). GRU has a relatively high score compared to LSTM (peephole).

In terms of the performance of BLEURT, the overall score is relatively low, which indicates that the learning ability of single-layer cyclic neural network is limited, but it is still obvious that the LSTM score is higher than other models.

It shows that adding peephole connections to LSTM is actually harmful to the model in the field of text generation. GRU is better than LSTM in many applications, but there is a significant gap between the ability of long text generation and LSTM in capturing complex semantics.

Experimental results show that the original LSTM has natural advantages in the field of long text. The improved version GRU made for LSTM and the added peep link have practical value in some fields, but the quality of text generated in the field of long text is obviously inferior to that of LSTM.

5. Summary

Text generation is the most representative application of NLP, but at present, text generation seems to have encountered a technical bottleneck. Besides news and other fact-based text with good effect, text generation, such as prose and poetry, is almost still in the primary stage. In 2018, the pre-training model bert brought NLP into a new stage, and achieved the leading achievement in many tasks. However, until today, the text generation has not achieved the ideal result, mainly because the pre-training model based on transformer model inefficiently handles the text with specific order and lacks the evaluation index of an effective language model to guide the optimization and improvement of the model. In this experiment, the performance of LSTM, GRU and LSTM with visual hole connection in long text generation is comprehensively compared with the latest evaluation index BERTscore and BLEURT. It is found that LSTM can have the highest quality of text generation. In addition, the pre-training model XLNet proves that the performance of LSTM in long distance dependence transformer is inferior to that of LSTM. Therefore, we can draw a relatively clear conclusion that LSTM in long text generation field has natural advantages. The future text generation model can be based on LSTM or refer to the structure idea of LSTM. The research conclusions and methods of this experiment are very practical. The strong recursion of LSTM is an inevitable idea of language model, and it is also an important guarantee for language model to generate strong logical text. The text fragments generated by language model are highly dependent on the text previously generated. The continuous improvement of similarity between new evaluation indexes and artificial evaluation directly affects the development and improvement of language model. At the same time, CBOW used in the experiment aims to improve the quality of text generation to the maximum extent from the angle of improving the quality of word vector and the angle of cluster search help prediction, which also shows that there are many ways and angles to improve the quality of text generation that we should think and try.

6. Closing

BERT marks the arrival of the pre-training model era. Its outstanding achievements have a positive impact on many fields. However, the ideas of large-scale text pre-training on the database, fine-tuning on specific data sets and bidirectional language model in BERT are clearly reflected in ELMo as early

ISSN: 2414-1895 DOI: 10.6919/ICJE.202105_7(5).0067

as a year ago. As an earlier pre-training model, ELMo also showed superior performance at that time. By using the series of left-to-right and right-to-left LSTM to generate downstream tasks, such bidirectional model has obvious effect. All these show that BERT has a lot of reference for ELMo. There is always a progressive or reference relationship between models at different times. Sometimes, it is necessary to understand the development of technology in an all-round way to truly understand its internal logic, so as to have better innovation. For example, the Transformer-XL used in the latest model XLNet, its core algorithm includes fragment recursion mechanism, which is to learn from the structure idea of RNN. In fact, the RNN structure has been added to self-attention in recent studies. Many researchers believe that transformer powerful feature extraction has made LSTM lose value in use, and that it is against the spirit of science to negate old things with new things. It is hoped that this research conclusion and scientific research methods can give some enlightenment and reference to other researchers in the field of text generation.

References

[1] Li Su. The market approach and reflection of the development of machine journalis- Taking Autamated Insights as an example[J]. press. 2015(18):56-61(in Chinese)

[2] Wan Xiaojun, Feng Yansong, Sun Weiwei. Research Progress and Trend of Automatic Text Generation. CCF Chinese Information Technology Professional Committee (in Chinese)

[3] Brown T B, Mann B, Ryder N, et al. Language Models are Few-Shot Learners[J]. 2020.

[4] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[5] Sak, H. & Senior, Andrew & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 338-342.

[6] Gers F A, Schmidhuber J. Recurrent nets that time and count[C]// Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. IEEE, 2000.

[7] Sun Baoshan,Li Wei.Research on Chinese Word Segmentation with Circulatory Network Connected by Peepholes[J]. Computer Engineering and Applications, 2019(19). (in Chinese)

[8] Cho, Kyunghyun & van Merriënboer, Bart & Gulcehre, Caglar & Bougares, Fethi & Schwenk, Holger & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. 10.3115/v1/D14-1179.

[9] Wiseman S, Rush A M. Sequence-to-Sequence Learning as Beam-Search Optimization[J]. 2016.

[10] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer ence, 2013.

[11] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2014.

[12] Zhang T, Kishore V, Wu F, et al. BERTScore: Evaluating Text Generation with BERT[J]. 2019.Zhang T, Kishore V, Wu F, et al. BERTScore: Evaluating Text Generation with BERT[J]. 2019.

[13] Sellam T, Das D, Parikh A P. BLEURT: Learning Robust Metrics for Text Generation[J]. 2020.

[14] Peters M, Neumann M, Iyyer M, et al. Deep Contextualized Word Representations[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018.

[15] Yang Z, Dai Z, Yang Y, et al. XLNet: Generalized Autoregressive Pretraining for Language Understanding [J]. 2019.

[16] Qin Y, Song D, Chen H, et al. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction[J]. 2017.