Research About Seq2seq and Transformer on Chinese Text Generation

Zhibo Liang

zbliang_sasee@buaa.edu.cn

Abstract. This paper performs a research about Chinese text generation. Firstly, the seq2seq model consisted of GRUs and Transformer model composed of attention mechanism are constructed. Then the given datasets are utilized to train the both two models and the parameters tuning is operated based on the training result. As a result, the test performance of text generation based on seq2seq and Transformer is acquired.

1 Introduction

Text generation is the process of using artificial intelligence to produce human-readable text. Text generation has become an increasingly significant area of artificial intelligence and natural language processing. It can be applied in many fields which conclude content creation assistance and conversational AI. For example, Text generation can augment human creativity and productivity by assisting with content creation tasks like article writing, story generation, product descriptions, and more. This helps save time and effort for human writers and content creators.

RNN (recurrent neural network) is considered a foundational text generation model that was introduced quite early in the history of the deep learning field. Hochreiter et al.[1] proposed LSTM (Long Short-Term Memory networks) in 1997 to solve the long-term dependencies problem. In 2014, the GRU (gated recurrent unit) was proposed by Cho et al.[2], which is another widely-watched gated network after LSTM. Sutskever et al.[3] apply encoder-decoder structure to sequence-to-sequence learning tasks just in the same year. And in 2017, Vaswani et al.[4] proposed Transformer, which is a network combined attention and greatly promoted the development of text generation.

As the underlying language models continue to advance,the applications and impact of text generation technology will likely grow considerably in the coming years. It's an area that holds great promise for transforming how we create, consume and interact with information and content. In this paper, a research about seq2seq and Transformer on Chinese text generation is conducted.

The structure of the paper: Sect. 2 introduces the methodology of this paper; Sect. 3 details the performance of simulation; Immediately afterwards, Sect. 4 summarizes.

2 Methodology

2.1 Seq2seq

Seq2seq is a machine learning model architecture for processing sequence data. It is widely used in natural language processing tasks such as machine translation, dialogue systems, text summarization, etc. The seq2seq model in this paper mainly concludes encoder-decoder and GRU structure. The detailed structure of this model is shown in Fig.1.

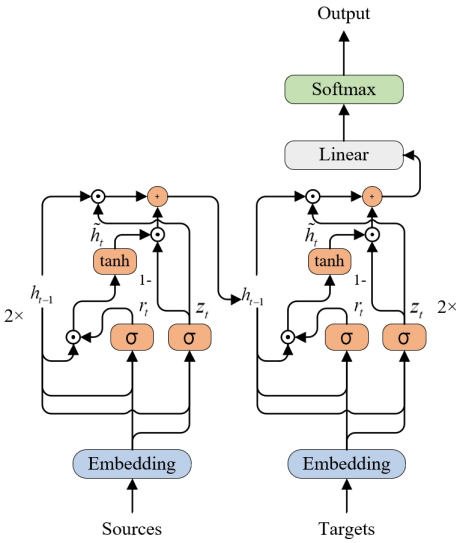

Fig. 1: The architecture of seq2seq model in this paper.

Encoder-decoder The encoder-decoder is an architecture that consists of two main components.

Encoder is the first part, which takes the input sequence and encodes it into a fixed-length vector representation, often referred to as the "context vector". We can use a function to express the transformation of encoder:

$$\boldsymbol{c} = f(\boldsymbol{h}_1, ..., \boldsymbol{h}_T) \tag{1}$$

where c represents context vector; $\boldsymbol{h}_t$ is hidden states of encoder.

Decoder is the second part, which takes the context vector from the encoder and generates the output sequence one token at a time. We can also use a function to express the transformation of the decoder's hidden layer:

$$\boldsymbol{s}_t = g(y_{t-1}, \boldsymbol{c}, \boldsymbol{s}_{t-1}) \tag{2}$$

where $\boldsymbol{s}_t$ represents hidden states of decoder; y_{t-1} is the previous step's target token.

The encoder and decoder are typically implemented using recurrent neural networks (RNNs), such as LSTMs or GRUs, which are well-suited for handling variable-length sequential data. The detailed structure of encoder-decoder is shown in Fig.2.

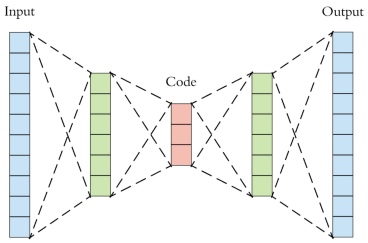

Fig. 2: The structure of encoder-decoder.

GRU The GRU offered a streamlined version of the LSTM memory cell that often achieves comparable performance but with the advantage of being faster to compute. As shown in Fig.3, GRU contains two gates: the reset gate and the update gate. These gates are given sigmoid activations, forcing their values to lie in the interval $(0, 1)$. Intuitively, the reset gate controls how much of the previous state we might still want to remember. Likewise, an update gate would allow us to control how much of the new state is just a copy of the old one. Suppose that the input is a minibatch $\boldsymbol{x}_t \in \mathbb{R}^{n\times d}$ and the hidden state of the previous time step is $\boldsymbol{h}_{t-1} \in \mathbb{R}^{n\times h}$. Then the reset gate $\boldsymbol{r}_t \in \mathbb{R}^{n\times h}$ and update gate $\boldsymbol{z}_t \in \mathbb{R}^{n\times h}$ can be computed as follows:

$$\boldsymbol{r}_t = \sigma(\boldsymbol{x}_t\boldsymbol{W}_{xr} + \boldsymbol{h}_{t-1}\boldsymbol{W}_{hr} + \boldsymbol{b}_r) \tag{3}$$

$$\boldsymbol{z}_t = \sigma(\boldsymbol{x}_t\boldsymbol{W}_{xz} + \boldsymbol{h}_{t-1}\boldsymbol{W}_{hz} + \boldsymbol{b}_z) \tag{4}$$

Next, we utilize reset gate $\boldsymbol{r}_t$ to calculate candidate hidden state $\widetilde{\boldsymbol{h}}_t \in \mathbb{R}^{n\times h}$ at time step t:

$$\widetilde{\boldsymbol{h}}_t = tanh(\boldsymbol{x}_t\boldsymbol{W}_{xh} + (\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1})\boldsymbol{W}_{hh} + \boldsymbol{b}_h) \tag{5}$$

where the symbol $\odot$ is the Hadamard product operator.

Finally, the new hidden state $\boldsymbol{h}_t \in \mathbb{R}^{n\times h}$ can be computed by the final update equation for the GRU:

$$\boldsymbol{h}_t = \boldsymbol{z}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{z}_t) \odot \widetilde{\boldsymbol{h}}_t \tag{6}$$

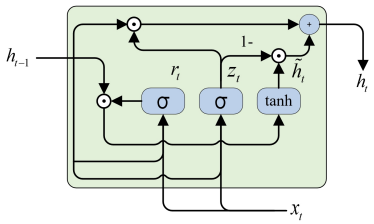

Fig. 3: The architecture of GRU.

2.2 Transformer

As an instance of the encoder–decoder architecture, the overall architecture of the Transformer is presented in Fig.4. As we can see, the Transformer is com-

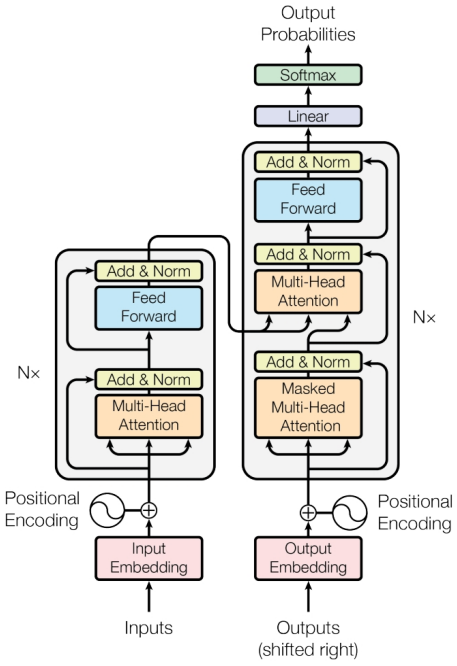

Fig. 4: The architecture of Transformer[4].

posed of an encoder and a decoder. In contrast to common sequence-to-sequence learning, the input (source) and output (target) sequence embeddings are added with positional encoding before being fed into the encoder and the decoder that stack modules based on self-attention.

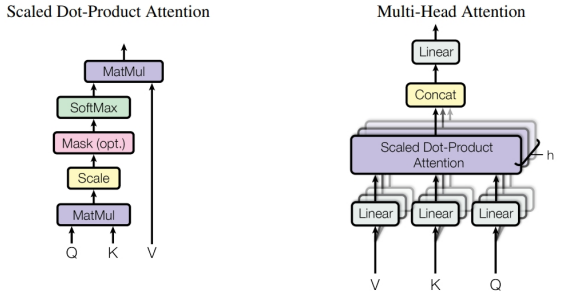

Fig. 5: (left) Scaled Dot-Product Attention, (right) Multi-Head Attention[4].

Attention An attention function can be described as mapping a query and a set of key-value pairs to an output,where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

The attention applied in this paper is Scaled Dot-Product Attention (Fig.5). The input consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values. In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix $\boldsymbol{Q}$. The keys and values are also packed together into matrices $\boldsymbol{K}$ and $\boldsymbol{V}$. We compute the matrix of outputs as:

$$Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}})\boldsymbol{V} \tag{7}$$

Instead of performing a single attention function, Multi-Head Attention (Fig.5) is applied in Transformer. Multi-Head Attention allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

$$MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, \ldots, head_h)\boldsymbol{W}^O \tag{8}$$

where $\boldsymbol{W}^O \in \mathbb{R}^{hd_v \times d_{model}}$ is the trainable weight matrix; d_{model} is the embeddings dimension.

Positional Encoding Since Transformer contains no recurrence and no convolution, in order for the model to make use of the order of the sequence, we must inject some information about the relative or absolute position of the tokens in the sequence. To this end, we add "positional encodings" to the input embeddings at the bottoms of the encoder and decoder stacks. The positional encodings have the same dimension as the embeddings, so that the two can be summed. In this paper, position encoding is operated as follows:

$$P_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \tag{9}$$

$$P_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \tag{10}$$

where pos is the position and i is the dimension.

With attention and position encoding, Transformer has unique advantages. For example, the structure of recurrence can only perform serial computations over time, and as the sequence length increases, the computational resource requirements will become higher and higher. Convolutional structures have properties of spatial invariance and locality, but are not well-suited for learning long sequences. In contrast to them, Transformer is able to effectively handle long sequence models, while also achieving parallel computation over time.

3 Experiment Results

3.1 Seq2seq

In this seq2seq model, the encoder and decoder both use two-layer GRUs with 32 hidden units. The loss function is selected to be cross-entropy loss; learning rate is set to 0.005; optimizer is Adam.

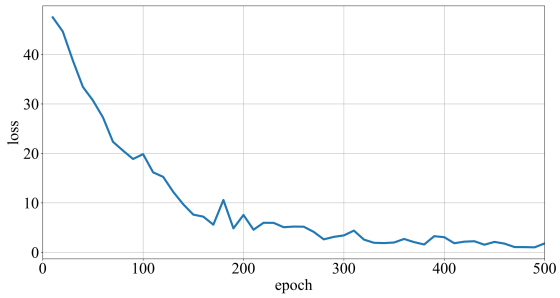

Fig. 6: The training result of seq2seq.

Input tokens	Correct tokens	Generated tokens
又称文莱渤泥婆罗乃	文莱以及英语 Brunei 均系同一地名之音译虽和中土相隔海程万里但向来仰慕中华宋朝太平兴国二年其王向打即苏丹中国史书上译为向打曾遣使来朝进贡龙脑象牙檀香等物其后朝贡不	文莱以及英语 Brunei 均系同一地名之音译虽和中土相隔海程万里但向来仰慕中华宋朝太平兴国二年其王向打即苏丹中国史书上译为和广东愈甚是吆喝着孙仲寿上来之中窥探上上
路不拾遗夜不闭户人人讲信修睦仁义	和爱今日眼见却是大不尽然还远不如渤泥国蛮夷之地感叹了一会就倒在床上睡了刚蒙胧合眼忽听见门外犬吠之声大作跟着有人怒喝叫骂蓬蓬蓬的猛力打门老婆婆下床来要去开门老头儿摇手止住轻轻对	和爱今日眼见却是大不尽然还远不如渤泥国蛮夷之地感叹了一会就倒在床上睡了刚蒙胧合眼忽听见门外犬吠之声大作跟着有人怒喝叫骂蓬蓬蓬的猛力打门老婆婆下床来要去开门老头儿摇手止住轻轻对

Fig. 7: The testing result of seq2seq.

Our practical validation revealed that the seq2seq model is not well-suited for processing very long time step sequences, so the number of input time steps was chosen to be 5. Fig.6 shows the training result of seq2seq in 500 epochs.

The testing result, based on the seq2seq trained before, which selects the words with the highest probability, is shown in Fig.7.

3.2 Transformer

In this Transformer model, the dimension of hidden layers is 32; both the Transformer encoder and the Transformer decoder have two layers using 4-head attention.The loss function is selected to be cross-entropy loss; learning rate is set to 0.005; optimizer is Adam. Different from seq2seq, Transformer does well in processing long time step sequences, so the number of input time steps was selected to be 20. Fig.8 shows the training result of Transformer in 500 epochs.

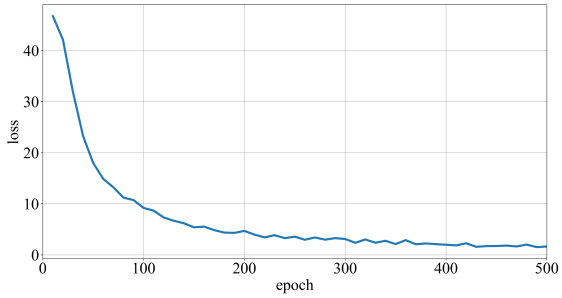

Fig. 8: The training result of Transformer.

After getting the trained Transformer model, we perform the testing and acquire the result as Fig.9.

Input tokens	Correct tokens	Generated tokens
又称文莱渤泥婆罗乃文莱以及英语 Brunei 均系同一地名之音译虽	和中土相隔海程万里但向来仰慕中华宋朝太平兴国二年其王向打即苏丹中国史书上译为向打曾遣使来朝进贡龙脑象牙檀香等物其后朝贡不绝麻那惹加那	又和中土相隔海程万里但向来仰慕中华宋朝太平兴国二年其王向打即苏丹中国史书上译为向打曾遣使来朝进贡龙脑象牙檀香等物其后朝贡不绝麻那乃雇船
路不拾遗夜不闭户人人讲信修睦仁义和爱今日眼见却是大不尽然还远不如渤泥国蛮夷之地感叹了一会	就倒在床上睡了刚蒙胧合眼忽听见门外犬吠之声大作跟着有人怒喝叫骂蓬蓬蓬的猛力打门老婆婆下床来要去开门老头儿摇手止住轻轻对张朝唐道相公你到	就倒在床上睡了刚蒙胧合眼忽听见门外犬吠之声大作跟着有人怒喝叫骂蓬蓬蓬的猛力打门老婆婆下床来左手持钢叉右手提着黄黑相间的坐满了晚饭过后杨鹏举

Fig. 9: The testing result of Transformer.

3.3 Evaluation

From the above result, it is obvious that both seq2seq and Transformer have achieved relatively good performance, but they still have some difference. In detail, the train loss of Transformer is a little lower than seq2seq, which demonstrate that the learning ability of Transformer is higher because of attention mechanism. But when the testing result is considered, Transformer does not show a significant advantage because the datasets are not large enough. In other words, overfitting occurs. In small datasets scenarios, the training of attention mechanisms is more difficult, because their correlations are more complex. It can be acquired from the testing result that when the semantic distance is relatively long, Transformer's performance is better than seq2seq. On the contrary, Transformer's performance is worse than seq2seq when the semantic distance is short.

4 Conclusion

This paper conducted a study on text generation based on seq2seq and Transformer. The training and testing results show that both two models can get excellent performance. But the learning ability of Transformer is obviously stronger than seq2seq because of the lower train loss. Small datasets applied in this paper limit the capability of Transformer, so research on larger datasets will be performed in the future.

References

1. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
2. K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
3. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.
4. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.