Sentiment Analysis based on Transformer with Only Encoder

Zhenyu Xie*

North China Electric Power University, Baoding 071003, China

Abstract

This paper studies how to use the transformer to solve the sentiment analysis problem. By removing the decoder and only retaining the compiler, the transformer is more suitable for handling this type of task. Introduced in detail the processing flow of the yelp comment data collection, and through various comparisons with LSTM and 1-D CNN, it is concluded that transformer with only encoder is better for long sequence tasks.

Keywords

Transformer; LSTM; 1-D CNN; Sentiment Analysis; Encoder.

1. Introduction

As social media becomes more and more close to life, people are used to express their opinions on social platforms such as Weibo and Twitter, and evaluate food on Meituan, Ele.me, Dianping, etc. A large number of easily accessible natural languages are on the Internet. It can be seen everywhere. Sentiment analysis of these texts also emerged and gradually demonstrated its irreplaceable role. Sentiment analysis shines in many scenarios. For example, for business operations, operators can use consumer feedback to adjust products and even their own development strategies in a timely manner. The main task of text sentiment analysis is to analyze sentimental text information, extract sentiment features from it, and make judgments on the features. For example, when judging sentences, they mainly focus on the acuteness of emotions, such as positive, negative, neutral, etc.

In recent years, sequence task modeling based on deep learning has become popular, and traditional classification, prediction, and generation tasks have been better solved in neural networks.Before Transformer [1] was proposed in 2017, the modeling of sequence modeling and sequence conduction (machine translation) tasks used more RNN based on encoder-decoder and attention mechanism (GMNT [2] etc. in 2016), CNN (ConvS2S[3] in 2017, Neural GPU[4] in 2016, ByteNet[5] in 2017, etc.) model. However, models based on RNN and CNN have obvious limitations. RNN cannot efficiently perform parallel computing, and the training period and reasoning delay are lengthy; the CNN model has limited ability to solve long-dependent problems.

The proposal of the Transformer model has brought revolutionary changes. The model abandons the traditional RNN and CNN structure and builds a network structure based only on the encoder-decoder module and attention mechanism. At the same time of efficient parallel training, the self-attention mechanism is used so that each position in the sequence can pay attention to the information of any position and build a dependency relationship. Although the Transformer is trained in parallel, it still needs to be predicted word by word in the prediction phase. Its accuracy increases with the deepening of the network, but too many parameters of the model limit the deepening of the network (high computational cost). The sequence length of the trained model is fixed. Once the sequence length exceeds the training sequence length, the sequence needs to be sliced, and the slice position is often not the boundary of the sentence, leading to context tearing. Although the model has many shortcomings, this creative idea has opened up a new way forward for later scholars. The literature based on Transformer has exploded in the past two years. The more representative one is the basic model: Transformer XL[6], Sparse Transformer[7]; Pre-training models: BERT[8], XLNet[9], ERNIE[10].

2. Methodology

2.1. Data Preprocessing

Along with the movement of the target, the sink node timely notifies the sensor nodes in the relevant detection area to join in the process of target tracking. Figure 1 is the flow chart of the moving target tracking process.

The dataset used in this article comes from the comment set in the Yelp dataset challenge. The data contains 600W yelp comments from different countries/regions, and the natural language is English. Packaged with the comment is the star information given by the user. According to the star ratings of user reviews, emotional tendencies are divided into five levels, 1, 2, 3, 4, and 5.

The Yelp data set consists of JSON objects, one JSON object per row, and the fields of the data set are as follows in Table. 1

Table 1. Detailed explanation of the JSON object fields of the review.json file of the Yelp datasetFigure

Field name	Example	effect
review_id	Q1sbwvVQXV2734tPgoKj4Q	Unique ID, length 22
user_id	hG7b0MtEbXx5QzbzE6C_VA	Unique ID, length 22
business_id	ujmEBvifdJM6h6RLv4wQIg	Unique ID, length 22
stars	1.0	Star rating, user rating
useful	6	useful votes received
cool	0	Cool votes received
funny	1	Interesting votes received
text	Total bill for this horrible service? Over $8Gs. These crooks actually had the nerve to charge us $69 for 3 pills.	Comment text
date	2013-05-07 04:34:36	Comment date

2.2. Create a Dictionary

In this experiment, after the stop words were deleted, the word segmentation method of morphological restoration was used to segment the comment text. From the word segmentation results, words with a frequency of more than 30 were extracted to make a dictionary. Introducing special tag <UNK> to refer to the word dictionary does not appear, and inserting a head of each sentence <S> tag, representing a sentence begins. After having a dictionary, it is necessary to establish a mapping relationship between words and numbers. In this experiment, the sequence number in the dictionary as a word, i.e., <UNK> corresponding to the number 0, <S> 1, the other words in ascending order. After the sentence is converted into a digital vector, the sentence is filtered according to maxlen (the maximum sequence length). When the sentence length is less than maxlen, the vector is filled with 0, and the sentence length is greater than maxlen is discarded. In this way, a data set with a length of maxlen is obtained, which provides convenience for parallel training.

2.3. Transformer with Only Encoder

Since the Transformer model was proposed in 17 years, it has quickly occupied the high ground of NLP tasks and broke the cognition of natural language processing, namely RNN. The BERT and GPT models that are currently widely used in natural language processing are based on this model.

The overall architecture of Transformer is shown in Figure. 1, including Inputs, Outputs, embedding, Positional Encoding, Encoder, Decoder, Softmax, and Output Probabilities. Inputs is the processed sentence vector, for example I am happy -> [10,15,698], the vector element represents the sequence number of the word at the corresponding position in the dictionary. Input Embedding is based on the word embedding representation obtained after the lookup table is checked by Inputs, as shown in Figure. 2

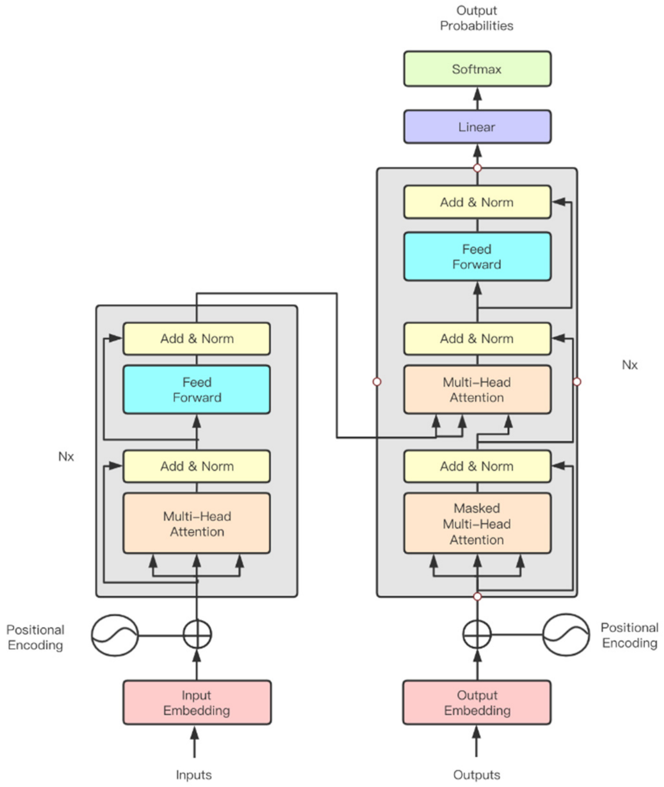

Figure 1. Transformer overall architecture

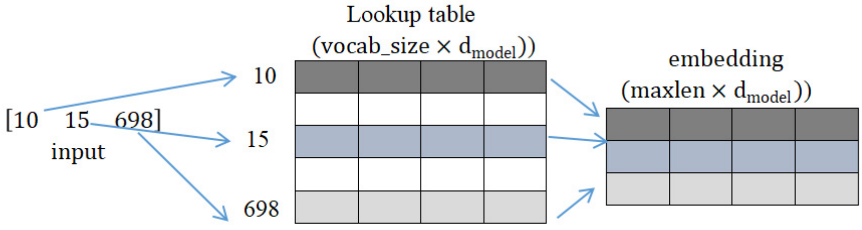

Figure 2. Get embedding through lookup table

Position coding is to add position information to the model, because the transformer is a parallel operation, which is different from the step by step of RNN which naturally has position information. When the transformer performs parallel calculation, if there is no position embedding, then the information at each position is The network is equivalent. The meanings of sentences composed of the same words in different orders are often different. For example, I love you scrambles the order of the subject and the object to get you love I, not to mention the grammatical problem, just from the meaning of the sentence In terms of, tremendous changes have taken place, and a transformer without position coding is unaware of this. Therefore, the introduction of position coding is necessary. The transformer uses trigonometric functions as the position coding, as shown in formulas 5-1 and 5-2, where pos represents the position in the sentence, and 2i and 2i+1 respectively represent the odd numbers in the vector Bits and even-numbered bits represent the length of embedding, and the length of position embedding is the same as that of embedding. The calculated position embedding and embedding are added to become the input of Multi-Head Attention.

The encoder-decoder structure is more suitable for the seq to seq model. The sentiment analysis task in this article only needs the encoder structure. The reconstructed network model is shown in Figure. 3. It is worth noting that in the output of the encoder layer, we only take the first output of the sequence (the output corresponding to the <S> tag) and pass it back.

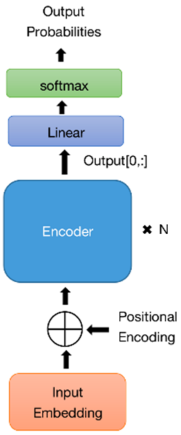

Figure 3. Transformer with only encoder

3. Results and Discussion

3.1. Loss Curves of Different Models under the Same Conditions

Let the latitude of the word vector=512, batchsize=64, learning rate lr=0.00001, the number of attention heads head=8, and the maximum sentence length maxlen=200.

The loss curve of the five-category Yelp training set is shown in Figure. 4. By comparing Table 2, The F1 score gap between models is also improved due to the increase in the difficulty of the task, and the transformer obviously has better processing capabilities. Followed by the LSTM model with Multi-Head Attention, it can be concluded that the advantages of the transformer can be better reflected when dealing with complex tasks.

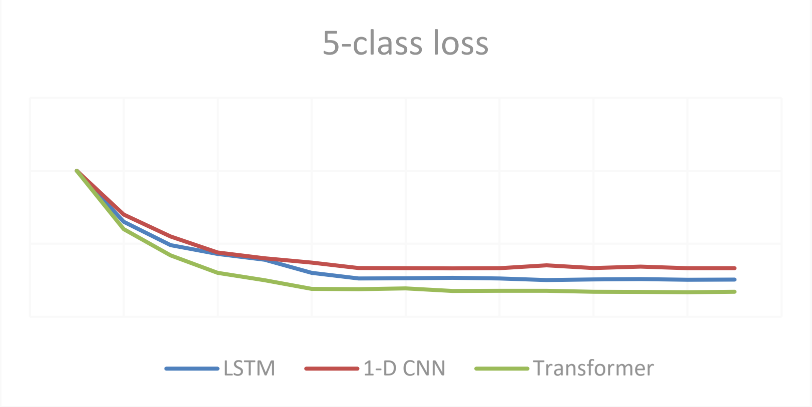

Figure 4. Five-category loss curves

Table 2. F1 value comparison

Model	5-Class
1-D CNN	0.658
LSTM	0.701
Transformer	**0.725**
LSTM(M-H A)	0.707

3.2. Long Dependency Test

The experiment extracts sentences with a sentence length of 201-400 from the comment data set as the long dependency data set. At this time, maxlen=400, and other conditions are consistent with Table 3. Transformer also has better accuracy on the problem of long dependence, and the result of LSTM is better than that of LSTM, indicating that the attention mechanism is beneficial to solving the problem of long dependence.

Table 3. Comparison of long-dependent F1 values

Model	5-Class
1-D CNN	0.614
LSTM	0.663
Transformer	0.702
LSTM(M-H A)	0.675

4. Conclusion

Transformer, which uses position coding to embed the position information of the sequence, can calculate in parallel while having position information, which greatly improves the training speed. Through the attention mechanism, it is easy to model the long dependence of the sequence, and the information will not decrease with the increase of the distance. In terms of calculation, the token information at a short distance is equivalent to the token information at a long distance. And Transformer is not perfect. The same as 1D-CNN is that Transformer is also a fixed-length sequence network, and the input sentence cannot exceed the maximum sequence length set during training, otherwise the sentence will be cut. In addition, the Transformer is calculated in parallel during training, but due to the design of the decoder, in the seq to seq reasoning, the results must be inferred according to the order step by step, but the transformer used in this article is the encoder transformer, so it cannot be very good. Show this shortcoming. Another big disadvantage of Transformer is that different encoder layers do not share weights, have a large amount of parameters, and require a lot of memory for inference. They are not suitable for the environment of low-end edge computing and need to be compressed.

References

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention Is All You Need[J]. 2017.

[2] Wu Y , Schuster M , Chen Z , et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation[J]. 2016.

[3] Gehring J , Auli M , Grangier D , et al. Convolutional Sequence to Sequence Learning[J]. 2017.

[4] Lukasz Kaiser and Samy Bengio. Can active memory replace attention? In Advances in Neural Information Processing Systems, (NIPS), 2016.

[5] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. arXiv preprint arXiv: 1610.10099v2,2017.

[6] Dai Z , Yang Z , Yang Y , et al. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context [J]. 2019.

[7] Child R , Gray S , Radford A , et al. Generating Long Sequences with Sparse Transformers[J]. 2019.

[8] Devlin J , Chang M W , Lee K , et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[J]. 2018.

[9] Zhilin Y, Zihang D, Yiming Y, et al XLNet: Generalized Autoregressive Pretraining for Language Understanding arXiv : 1906.08237v1 [cs.CL].

[10] Cheng Y , Wang D , Zhou P , et al. A Survey of Model Compression and Acceleration for Deep Neural Networks[J]. 2017.